TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

TIIA LEPPÄNEN

HYVIÄ KÄYTÄNTÖJÄ KÄYTTÄJÄKESKEISEN SUUNNITTELUN INTEGROIMISEKSI KETTERIIN OHJELMISTOKEHITYSPROJEKTEIHIN

Diplomityö

Tarkastaja: Prof. Kaisa Väänänen
Ohjaaja: TkT Kati Kuusinen

Aihe hyväksytty Tieto- ja sähkötekniikan tiedekuntaneuvoston kokouksessa 7. joulukuuta 2016

TIIVISTELMÄ

TIIA LEPPÄNEN: Hyviä käytäntöjä käyttäjäkeskeisen suunnittelun integroimiseksi ohjelmistokehitysprojekteihin
Tampereen teknillinen yliopisto
Diplomityö, 85 sivua, 3 liitesivua
Helmikuu 2017
Diplomi-insinöörin tutkinto
Tietotekniikan DI-tutkinto-ohjelma
Pääaine: Käytettävyys
Tarkastaja: Professori Kaisa Väänänen
Ohjaaja: Tekniikan tohtori Kati Kuusinen

Avainsanat: Käyttäjäkeskeisyys, suunnittelu, käyttäjäkokemus, ohjelmistokehitys, ketteryys, Scrum, Lean, Kanban

Diplomityön tavoitteena oli löytää hyviä käytäntöjä ja toimintatapoja, joilla yritykset integroivat käyttäjäkeskeisen suunnittelun mukaan ketterään ohjelmistokehitykseensä. Käytäntöjen ja toimintatapojen löytämisen tarkoituksena on helpottaa ja auttaa muita yrityksiä käyttäjäkeskeisen suunnittelun integroimisessa osaksi ketterää ohjelmistokehitystä, joka edelleen koetaan paikoin haastavaksi.

Diplomityö toteutettiin kirjallisuusselvityksen, haastattelujen ja verkkoaineistoanalyysin avulla. Kirjallisuusselvityksellä perehdyttiin aiheeseen liittyvään tutkimukseen käytännöistä ja toimintatavoista etenkin käyttäjäkeskeisen suunnittelutyön osalta. Haastatteluiden avulla kartoitettiin kuuden suomalaisen ohjelmistoyrityksen käytäntöjä ja toimintatapoja integrointihaasteiden ylittämiseksi, ja näkemyksiä, miten integrointia voisi edelleen parantaa. Verkkoaineistoanalyysin avulla laajennettiin näkemystä tämän hetken todellisista ja vaihtoehtoisista käytännöistä ja toimintatavoista. Verkkoaineistoanalyysi piti sisällään blogeja, verkkoartikkeleja ja verkossa käytyjä aiheeseen liittyviä ammattilaisten keskusteluja.

Kirjallisuuden, haastattelujen ja verkkolähteiden pohjalta laaditut hyvä toimintatavat ohjeistavat ja auttavat ohjelmistoyrityksiä muodostamaan integroituja tiimejä, tukemaan sujuvaa kommunikaatiota, hyväksymään epäonnistumisia, ja parantamaan ketterää prosessia käytännön tasolla. Suositellut käytännöt ja toimintaohjeet toimivat ohjeistuksena myös muille kuin ohjelmistoalan yrityksille tilanteissa, joissa organisaation halutaan muotoutuvan ketterämmäksi.

ABSTRACT

TIIA LEPPÄNEN: Good Practices to Integrate User-Centred Design into Agile Software Development Projects
Tampere University of Technology
Master of Science Thesis, 85 pages, 3 Appendix pages
February 2017
Master of Science in Technology
Master's Degree Programme in Information Technology
Major: Usability
Examiner: Professor Kaisa Väänänen
Instructor: Doctor of Science in Technology Kati Kuusinen

Keywords: user-centred, design, user experience, software development, Agile, Scrum, Lean, Kanban

The purpose of the thesis was to find good practices and approaches which software companies use to integrate user-centred design into agile software development. The reason to find these practices and approaches is to help and make it easier for companies to implement the integration which is still quite challenging.

The thesis was carried out by a literature review, interviews and web content analysis. The literature review was performed to familiarize with the research of the practices and approaches, especially relating to the user-centred design work. The integration and design practices to overcome challenges of the integration was surveyed by interviews. Practices for improvint the integration were also discovered in interviews. The web content analysis expanded the approach of current and alternative practices and approaches. The web content analysis included related blogs, articles and discussions by professionals in the field.

Good practices based on literature, interviews and online sources give instructions and help software companies to create well-integrated teams, to support fluent communication, to accept failures and to improve agile processes on a practical level. Good practices and guidelines act as a framework also for other than software companies where the organisation is aspired to be more agile.

ALKUSANAT

Haluan kiittää kaikki haastatteluihin osallistuneita suunnittelijoita, jotka tarjosivat kattavasti tietoa ja näkemyksiään ketterästä ohjelmistokehityksestä ja käyttäjäkeskeisestä suunnittelusta. Haluan myös kiittää alkuperäistä diplomityöohjaajaani tekniikan tohtori Kati Kuusista, sekä professori Kaisa Väänästä arvokkaista neuvoista ja ohjeista diplomityön osalta, sekä ymmärryksestä kiireistä aikatauluani ja elämäntilannettani kohtaan. Näiden avulla sain opintoni vietyä valmistumiseen asti.

Erityisesti haluan kiittää äitiäni, veljeäni sekä muuta perhettäni ja ystäviäni tuesta ja ymmärryksestä koko yliopisto-opintojeni aikana.

Espoossa, 13.12.2016

Tiia Leppänen

SISÄLLYSLUETTELO

LYHENTEET JA MERKINNÄT

Asiakaskokemus	*engl. Customer experience, CX*; Asiakkaan kohtaamiset, mielikuvat ja tunteet ollessaan vuorovaikutuksessa yrityksen ja sen tuotteen kanssa. Asiakaskokemuksen ymmärtäminen on olennainen osa asiakkuuden hallintaa. Asiakaskokemus heijastaa sitä, miten asiakas kokee yrityksen ja sen tarjonnan. Vapaa käännös ja yhdistelmä lähteistä [60] ja [48]
Back end	Sivuston taustaosat ja -toiminnot, eli kaikkea sitä toimintaa, mitä käyttäjä ei näe, kuten kääntämistä tai käyttöarkkitehtuurin optimointia. [49]
Front end	Selaimen tulkitsema ja ihmiskäyttäjille visuaalisessa muodossa esitettävä osuus sivustosta, kuten tekstisisältö, värit, kuvat, typografia, käyttöliittymäkomponentit. [49]
Full stack -kehittäjä	Kehittäjä, joka tuntee back end ja front end -tekniikat, ja kykenee toimimaan molempien kehittäjänä.
Hukka	*engl. Muda, waste*; Kaikki, mikä ei lisää arvoa lopputuotteeseen tai palveluun asiakkaan näkökulmasta. [55]
Imuohjaus	Tuotannonohjauksen muoto, jossa tuotteita valmistetaan vain, jos asiakkaat niitä tilaavat [55].
Integrointi	*engl. Integration*; Yhdentäminen, yhdentyminen, sisällyttäminen johonkin. [63]
Iso kuva	Tässä diplomityössä termillä viitataan projektin kokonaiskuvaan. Iso kuva pitää sisällään muun muassa vaatimusten ja toiminnallisuuksien keskinäiset riippuvuudet ja vaikutukset toisiinsa, jotka vaikuttavat tuotteen ja vision toteutukseen.
Iteraatio	*engl. Iteration*; Toistoa, looppi, luuppi. [63]
Jidoka	Käytetään myös termiä autonomaatio. Siinä inhimillinen äly yhdistetään koneeseen, joka pysähtyy, kun ongelma ilmenee. Jidoka viittaa kykyyn keskeyttää tuotanto, mikäli ilmenee toimintahäiriö, laatuvirhe tai muu vastaava. Jidoka auttaa estämään päästämästä virheellisiä suoritteita eteenpäin arvovirrassa, tunnistamaan ja ratkaisemaan ongelmia, ja rakentamaan laatua tuotantoprosessiin. [55]
Juuri oikeaan tarpeeseen (JOT)	*engl. Just in Time, JIT*; Antaa asiakkaille sitä mitä he haluavat silloin kun he sitä haluavat tietyn laatuisena hyödyntäen mahdollisimman vähän voimavaroja (työvoima, tila ja kesken olevat työt (WIP)). [55]

Kanban	Japaninkielinen termi, joka tarkoittaa signaalia. Kanban liittyy visuaalisiin työkaluihin, jotka kuvaavat jonkun tarkasteltavan asian todellista tilaa. Sitä käytetään imuohjausjärjestelmässä signaloimaan, koska tuotannon tulisi alkaa. Kanban voi saada eri muotoja (kortit, taulut, valot, korit jne. [55]
Käyttöliittymä	*engl. User Interface, UI*; Välineet ja toiminnot, joilla käyttäjä viestii tietojärjestelmän kanssa. [54] Tässä diplomityössä käyttöliittymällä tarkoitetaan näyttöpäätteellä olevaa käyttöliittymää, joka toimii rajapintana sovelluksen käytölle.
Läpimenoaika	*engl. Lead time*; Tarvittava aika tilauksen vastaanottamisesta tuotteen toimittamiseen asiakkaalle. [55]
MVP	Minimum Viable Product, pienin mahdollinen toteutus tuotteesta.
Pullonkaula	Prosessin hitain vaihe, joka hidastaa koko prosessin etenemistä. [55]
Sprintti	(*engl. sprint*) Sprintin aikana tuotetaan ”valmiin” määritelmän täyttävä, käyttökelpoinen ja potentiaalisesti julkaisukelpoinen tuoteversio. Sprintti kestää enimmillään neljä viikkoa. Sprintit suoritetaan toinen toistensa perään. [89] Scrumissa termiä käytetään *iteraatio*n sijasta.
Startup	Nuori, kasvuhakuinen yritys.
Suunnittelija	Tässä diplomityössä termillä viitataan asiantuntijaan, joka tekee käyttäjäkeskeistä suunnittelua.
Suunnittelu	Tässä diplomityössä suunnittelulla tai suunnittelutyöllä tarkoitetaan käyttäjäkeskeiseen suunnitteluun liittyvän työhön.
Tuottavuus	Tuotetun hyödyn suhde panoksiin. [55]
Työpaja	*engl. Workshop*; Henkisen työn tekijöiden, taiteenharjoittajien tms. yhteinen työtila; yhdessä (sellaiseen) työskentelemään kokoontuneesta ryhmästä; em. ryhmän työskentelystä. [54]
Virtaus	Tuotantofilosofia, joka perustuu tuotteen siirtymiseen yhdeltä työpisteeltä toiselle yksi kerrallaan ilman, että väliin syntyy välivarastoa. [55]
WIP	Work In Progress, kehitteillä oleva työ.

.

1. JOHDANTO

Opiskeluaikana toimin Ohjelmistotekniikan laitoksella sekä Ihmiskeskeisen teknologian yksikön opintojaksoilla assistenttina. Tätä kautta sain hyvät perusteet ymmärtää sekä ihmiskeskeisen suunnittelun iteratiivista mallia, että ohjelmistotuotannon erilaisia läpivientimenetelmiä. Opintojen loppupuolella siirryin erääseen kansainväliseen ohjelmistoyritykseen, jossa projektien läpivientimenetelmänä käytettiin ketteriä menetelmiä. Menetelmiä käytettiin lähes kirjaoppimaisesti, sillä yritys edelleen opetteli ketteryyttä. Vaihdoin työpaikkaa yritykseen, joka on keskittynyt käyttäjäkeskeiseen suunnitteluun, ja toimi osana projekteja, joissa kehittäjät tulivat toisesta yrityksestä. Huomasin monia ongelmakohtia pyrittäessä läpiviemään ketteriä projekteja oppikirjan mukaisesti. Tässäkin asiakasyrityksessä edelleen opeteltiin toimimaan ketterästi, joka johti oppikirjamaiseen jäykkyyteen ketteryyden sijasta. Projekteissa ongelmien mittakaava oli valtava, ja vaati jatkuvia korjausliikkeitä, eikä varsinaiselle arvoa tuottavalle suunnittelulle ollut lopulta riittävästi aikaa. Aloin etsiä ratkaisuja tieteellisistä artikkeleista, blogeista ja alan kirjoista. Tuntui, ettei varsinaista ratkaisua tilanteeseen löytynyt, mutta vertaistukea kyllä. Tällöin päädyin kasvokkain sen kanssa, että mahdollisesti yleispätevää ratkaisua ei olisikaan. Tästä huolimatta halusin löytää ratkaisuja ongelmiin, joiden kanssa monet yritykset edelleen painivat. Näistä haasteista syntyi ajatus diplomityön aiheesta. Olin motivoitunut löytämään sen ohuen punaisen langan, jota yritykset voisivat seurata tehdessään ketteriä projekteja.

1.1 Tutkimuksen tausta

Tuotekehityksessä on alettu näkemään se arvo, jota käytettävyys- ja suunnittelutyö tuovat asiakassuhteelle ja lopputuotteelle. Suunnittelutyön, joka pitää sisällään niin konseptointia, käyttäjäkokemussuunnittelua kuin käyttöliittymäsuunnittelua ja testausta, integroiminen ohjelmistokehitysprosessiin silti kangertelee. 2000-luvun alussa yritykset alkoivat ottaa käyttöön vesiputousmallin sijasta ketteriä ohjelmistokehitysmenetelmiä. Moderni ohjelmistokehitys usein seuraa ketteriä menetelmiä [11]. Tarkoituksena muutoksessa ketteriin menetelmiin oli tehdä prosessista nopeampi, joustavampi ja säästää resursseja sekä keskittyä käyttäjän kannalta olennaiseen karsimalla ylimääräiset toiminnallisuudet ja vaatimukset. Nämä perusteet menetelmämuutokselle tukevat ajatusta, jota käyttäjäkeskeisessä suunnittelussa on korostettu käyttäjä- ja asiakaskeskeisyyttä. Miten sitten on mahdollista, että kaksi samaan päämäärään pyrkivää menetelmää integroituu niin huonosti toisiinsa? Miten yritykset ovat ratkaisseet integroimisongelmat?

Työryhmässä, joka julkaisi Ketterän ohjelmistokehityksen julistuksen vuonna 2001 [64], ei ollut käytettävyys- ja suunnittelupuolen edustajaa, eivätkä käytettävyysammattilaiset ole kokoontuneet yhteen ja puineet julistuksen arvoja ja periaatteita käyttäjäkeskeisen suunnittelun näkökulmasta. Tästä syystä ei ole olemassa standardeja, joiden mukaan käytettävyystyön integroiminen otettaisiin huomioon ketterässä ohjelmistokehityksessä. [7]

Käyttäjäkeskeisen suunnittelun ja ketterien ohjelmistokehitysmenetelmien integroimisessa yritykset ovat kohdanneet käytännön haasteita. Ketterät menetelmät keskittyvät ennen kaikkea prosessinäkökulmaan, eivätkä niinkään käytännön tekijöihin suunnittelijoiden ja kehittäjien näkökulmasta [40]. Iso haaste on yhdistää nämä kaksi iteratiivista menetelmää joustavasti toisiinsa. Esimerkiksi aikataulullinen sovittaminen yhteen on haastavaa, koska käyttäjäkeskeistä suunnittelua joudutaan tekemään etupainotteisesti usein liian pienin resurssein, kun taas muu kehitystiimi keskittyy vielä edellisen suunnittelukierroksen toteuttamiseen. Kehittäjät ja suunnittelijat painottavat työskentelyn eri iteraatioihin, ja silti heidän pitäisi tehdä työtä rinnakkain.

Aihe puhuttaa niin tieteellisissä artikkeleissa, käytettävyys- ja Agile-konferensseissa kuin erilaisissa blogi- ja nettikirjoituksissa. Aiheen ympärillä on tehty paljon tutkimusta, mutta niissä on pitkälti keskitytty ongelmiin, joita erikokoiset ja eri tahoille erikoistuneet yritykset ovat kohdanneet. Tutkimuksissa on perehdytty myös ongelmien syihin, ongelmien esiintyvyyden laajuuteen ja siihen, mihin kaikkiin ongelmat integroidussa käytettävyyden ja ketterien ohjelmistokehityksen prosessissa nämä vaikuttavat. Medioissa nostetaan esille ajatuksia ja ideoita, joilla intergroinnin haasteita voisi helpottaa, ja miten menetelmiä tulisi painottaa ja iteraatioida organisoida. Ratkaisut eivät kuitenkaan ole yleismaailmallisia, sillä jokainen yritys määrittelee ja sopeuttaa niin ketterät menetelmät kuin käytettävyyssuunnittelumenetelmät omaan organisaatioon sopivaksi, ja lisäksi niitä hiotaan ja sisältöjä painotetaan projektista ja asiakkaasta riippuen eri tavoin. Yleismaailmallisen menetelmän tulisi ottaa huomioon niin käyttäjäkeskeisen suunnittelun tarpeet, asiakkaan tarpeet, sidosryhmien tarpeet, kehitystiimin tarpeet, sekä projektinhallinnalliset ja markkinoinnilliset tarpeet. Käytännössä yleismaailmallista ohjeistusta näiden menetelmien integroimiseen on mahdotonta tehdä.

Tässä diplomityössä on haluttu painottaa niitä ratkaisukeinoja, joilla haasteista on päästy yli, ja miten yritykset ovat painottaneet omaa työtään ja toimintatapojaan saadakseen integraation menetelmien välillä toimimaan. Diplomityössä on keskitytty ohjelmistoyrityksiin, joissa on sekä käyttäjäkokemuksen suunnittelua, että varsinaista kehitystyötä.

1.2 Tutkimuksen tavoitteet, tutkimuskysymykset ja aiheen rajaus

Diplomityön tavoitteena on ollut selvittää menetelmiä ja käytäntöjä, joita ohjelmistoyritykset, joilla on myös käyttäjäkeskeistä suunnitteluosaamista, käyttävät ketterien menetelmien ja suunnittelumenetelmien yhdistämisessä. Tavoitteena on ollut löytää erityisesti ne suunnittelumenetelmät ja suunnittelutehtävät, jotka koetaan tuottavan arvoa koko läpimenoprosessille, ja ennen kaikkea lopputuotteelle, niin ettei prosessissa tuoteta hukkaa suunnittelijoille, kehittäjille, eikä asiakkaille.

Tutkimuskysymyksiä on asetettu kolme:

1. Miten käyttäjäkeskeinen suunnittelu otetaan huomioon osana ketterää ohjelmistokehitysprosessia?
2. Mitkä käytännöt tukevat suunnittelutyön integroitumista ketterään ohjelmistoprosessiin?
3. Mitkä suunnittelutehtävät koetaan tuottavan eniten hyötyä prosessin ja lopputuloksen kannalta?

Diplomityö on rajattu käsittelemään ketteristä menetelmistä ainoastaan Scrumia (kappale 3.1) ja Leania (kappale 3.2). Nämä kaksi menetelmää ovat Suomessa sovelletuimmat ohjelmistoyritysten puolella. Empiirisessä osassa tutkittavat ohjelmistoyritykset on rajattu kuuteen suomalaiseen melko nuoreen yritykseen, jotka ovat ottaneet käyttöön ketterät menetelmät tuotantoprosessissaan. Kaikilla tutkittavilla ohjelmistoyrityksillä on sekä kehittäjiä, eli ohjelmoijia, että suunnittelijoita, joiden osaaminen vaihtelee konseptoinnista ja käyttäjäkokemuksen suunnittelusta graafiseen ja visuaaliseen suunnitteluun.

Tutkimustavoitteisiin, tutkimuskysymyksiin ja aiheen rajaukseen palataan tarkemmin luvussa 5. Tutkimuskysymyksiin vastataan luvussa 8.

1.3 Tutkimusmenetelmät ja –prosessi

Työ koostuu teoreettisesta kirjallisuustutkimuksesta ja empiirisestä tutkimusosuudesta. Teoreettisessa osuudessa käsitellään teorioita, jotka tukevat empiirisen osuuden ymmärtämistä ja kehittämistä. Teoriaosuudessa tutustutaan aiheeseen liittyvään olemassa olevaan kirjallisuuteen ja tutkimukseen.

Työn empiirisessä osuudessa on otettu mukaan kuusi ohjelmistoyritystä, jotka tekevät käyttäjäkeskeistä suunnittelua osana ohjelmistoprojektia. Käyttäjiä on haastateltu yksilö- ja ryhmähaastatteluissa. Heidän avullaan olen selvittänyt olemassa olevia käytäntöjä, miten käyttäjäkeskeinen suunnittelu on saatu onnistuneesti yhdistettyä ketteriin ohjelmistoprojekteihin, ja millaisia toimintatapoja onnistuminen on edellyttänyt.

Empiirisessä osuudessa haastattelujen lisäksi on tutkittu verkkoaineistoja haastattelutulosten tueksi ja haastajaksi. Verkkoaineistoanalyysi tarjoaa lisää näkökulmia tämän hetken käytäntöihin ja toimintatapoihin, jotta integraatio saadaan onnistumaan.

Diplomityöprosessi alkoi teorian tutkimisella, ja teoriaosuuden kirjoittamisella. Empiirinen tutkimusprosessi alkoi kuuden käyttäjähaastattelun tekemisellä. Haastattelut nauhoitettiin ääninauhurilla, jotka kuuntelin läpi uudelleen haastatteluanalyysin tekoa varten. Haastatteluanalyysin jälkeen otin mukaan prosessiin verkkoaineistoanalyysin. Haastattelut antoivat pohjaa verkkoaineiston etsintään, sillä ne toivat esiin asioita ja näkökulmia, jotka muutoin olisivat jääneet verkkoaineistosta pois.

1.4 Diplomityön rakenne

Diplomityön teoriaosuudessa, luvuissa 2-4, käsitellään aluksi peruskäsitteet ja -menetelmät käyttäjäkeskeisestä suunnittelusta ja ketterästä ohjelmistokehityksestä. Teoriaosuuden lopuksi käydään läpi käyttäjäkeskeisen suunnittelun integroimista ketteriin ohjelmistokehitysprosesseihin eri kirjallisuuslähteiden pohjalta.

Empiirisessä osuudessa, luvussa 5, käydään läpi tutkimusmenetelmät, tutkimusasetelma sekä tutkimuksen eteneminen, ja saadut tutkimustulokset. Empiirisessä osassa on yritys-haastatteluiden lisäksi tutkittu eri verkkoaineistojen, kuten nettiartikkelien ja blogien, näkökulmia tutkimusaiheeseen. Näiden tuloksia on koottu yhteen haastattelutulosten kanssa luvussa 7.

2. KÄYTTÄJÄKESKEINEN SUUNNITTELU

Käyttäjäkeskeinen suunnittelu on oleellinen osa hyvän lopputuotteen suunnittelua. Se pitää sisällään konseptointia, käyttäjien ja asiakkaiden tarpeiden ja vaatimusten kartoitusta, käyttäjäkokemuksen suunnittelua, käyttöliittymäsuunnittelua, visuaalista suunnittelua, käyttäjä- ja käytettävyystestausta, sekä ratkaisujen toiminnallista määrittelyä. Näiden tehtävien avulla varmistetaan, että käyttäjät osaavat käyttää tuotetta oikein, tehokkaasti ja tarkoituksenmukaisesti. Edellisten lisäksi loppukäyttäjä kokee tuotteen käytön myös miellyttäväksi, viihdyttäväksi ja helppokäyttöiseksi sen tarkoitukseen nähden. Kun käyttäjää osallistetaan tuotteen suunnitteluun, voidaan varmistua, että tuote palvelee käyttäjiä edellisten määreiden mukaisesti. Tyytyväisestä asiakkaasta tulee lojaali, mikä tarkoittaa, että käyttäjäkeskeisellä suunnittelulla on iso ja tärkeä rooli yrityksen liiketoiminnassa. Liiketoiminnallisia päätöksiä voidaan perustella tiedoilla, joita on hankittu käyttäjäkeskeisen suunnittelun menetelmin käyttäjiltä ja sidosryhmiltä.

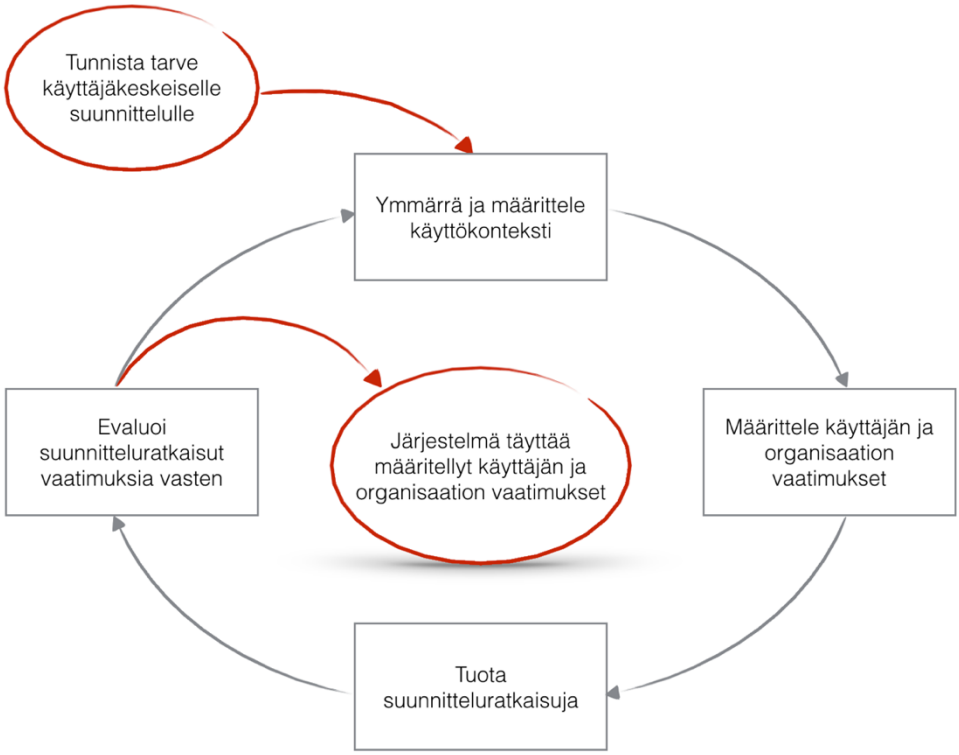
Käytettävyys (*engl. usability*) on käyttäjäkeskeisen suunnittelun (*engl. User-Centered Design, UCD*) peruskäsite, ja se standardisoitiin standardiin *ISO 9241-11* [2] vuonna 1998. Standardissa käytettävyys on määritelty seuraavasti: “Se vaikuttavuus, tehokkuus ja tyytyväisyys, jolla tietyt määritellyt käyttäjät saavuttavat määritellyt tavoitteet tietyssä ympäristössä.” Standardin mukaista määritelmää on kuitenkin kritisoitu sen suppeuden vuoksi. Käytettävyys-termiä käytetään monella tapaa, mikä on johtanut hämmentävään konseptiin, jossa eri näkökulmat ovat itse määritelleet käsitteen uudelleen. Konseptin ymmärtäminen kuitenkin vaatii vakiintunutta terminologiaa [37]. Green ja Pearson [16] kritisoivat myös vakiintuneen terminologian puutetta, ja sitä, että standardi keskittyy liiaksi tarkasti määriteltyihin tehtäviin ja tavoitteisiin, eikä siinä huomioida käyttäjäkokemusta. Standardissa painotetaan tärkeimmiksi ominaisuuksiksi tehokkuutta ja toimivuutta, mutta kaikkien tuotteiden osalta nämä eivät ole käyttäjälle tärkeimpiä ominaisuuksia. Tällaisia ovat esimerkiksi emotionaaliset tuotteet, joiden käyttäjäkokemusta on vaikea mitata. Käyttäjäkokemuksen määritelmä standardoitiin, *ISO 9241-210* [1], lopulta vuonna 2010.

2.1 Peruskäsitteet

Käyttäjäkeskeisellä suunnittelulla (*engl. Human-Centered Design, HCD*) tarkoitetaan prosessia ja menetelmiä, joita hyödynnetään ohjelmistoprojektin aikana, kun halutaan tuoda suunnitteluun mukaan käyttäjäkeskeinen näkökulma. Käyttäjillä viitataan yksilöihin, ovat valmiin tuotteen kanssa vuorovaikutuksessa. Käyttäjäkeskeisen suunnittelun

lähtökohtana toimii ISO 13407 –standardi vuodelta 1999 [3], jossa kuvataan käytettävyys periaatteiden, suunnittelun ja toimintojen kautta.

Alla olevassa kuvassa 1 on standardiin pohjautuva käyttäjäkeskeinen suunnitteluprosessi, ja sen iteratiivisuus. Prosessi lähtee liikkeelle käyttäjän tarpeiden tai toiveiden tunnistamisesta. Suunnittelijoiden tehtävänä on eri menetelmien avulla ymmärtää tarpeiden taustat, jotta tarpeesta muodostuu kokonaiskuva. Tämän jälkeen suunnittelija määrittelee käyttökontekstin. Seuraavaksi määritellään vaatimukset perustuen loppukäyttäjien tarpeisiin ja odotuksiin. Määrittelyjen jälkeen luodaan suunnitteluratkaisuja, joita evaluoidaan vaatimuksia vasten. Jos suunnitteluratkaisut ja vaatimukset kohtaavat, voidaan toiminnallisuutta lähteä toteuttamaan. Jos vaatimukset eivät täyty, suunnitteluratkaisuja iteroidaan, kunnes vaatimukset täyttyvät. Tässä tarvitaan yhteistyötä käyttäjien kanssa, jotta ratkaisuista saadaan palautetta.



Kuva 1 *Käyttäjäkeskeinen suunnitteluprosessi. [3]*

ISO 13407 -standardin mukaan käytettävyys muodostuu siitä, kuinka hyvin käyttäjät voivat käyttää tuotetta määritellyissä käyttötilanteissa, ja samalla saavuttaa määritellyt tavoitteet tehokkaasti ja tyytyväisinä. Standardi ei ota tarkemmin kantaa, mitä työkaluja ja menetelmiä suunnittelijoiden tulee käyttää tavoitteisiin päästäkseen. Kappaleessa 2.2 on kuvattu erilaisia menetelmiä, joita voidaan käyttää käyttäjäkeskeisen suunnitteluprosessin aikana tukemaan tavoitteiden saavuttamista.

Uusi käyttäjäkeskeisen suunnitteluprosessin standardi, *ISO 9241-210*, julkaistiin vuonna 2010 [1], ja se korvasi ISO 13407 -standardin [3]. Uusi standardi käsittelee vuorovaikutteisten järjestelmien käyttäjäkeskeisyyttä. Tässä uudessa standardissa selvennettiin iteraatioiden roolia koko suunnitteluprosessin ajan, ja korostettiin jatkuvaa ihmiskeskeisten suunnittelumenetelmien käyttöä koko järjestelmän elinkaaren ajan. ISO 13407 -standardin [3] tarkoituksena oli myös selventää ihmiskeskeisen suunnittelun periaatteita. Samalla standardi otti kantaa jo aiemmin laajalti käytössä olleeseen käyttäjäkokemus-termiin (*engl. User Experience, UX*) ja määritteli sen.

Käyttäjäkokemus liittyy ihmisen ja tietokoneen väliseen vuorovaikutukseen (*engl. Human-Computer Interaction, HCI*), jossa on myös emotionaalinen näkökulma. Tämän ISO 9241-210 -standardin [1] mukaan käyttäjäkokemus sisältää henkilön käsitykset ja reaktiot, jotka johtuvat käytöstä ja/tai odotettavissa olevasta tuotteen, järjestelmän tai palvelun käytöstä. Tämä koostuu käyttäjän tunteista, uskomuksista, mieltymyksistä, fyysisistä ja psyykkisistä vasteista, käyttäytymisestä ja aikaansaannoksista, jotka syntyvät jo ennen käyttöä, sen aikana, mutta myös sen jälkeen. Käyttäjäkokemus voidaan jakaa nautinnollisiin ja käytännöllisiin tekijöihin. Nautinnollisia ovat muun muassa käyttöliittymän ulkonäkö, houkuttelevuus ja hauskuus. Käytännöllisiä tekijöitä ovat muun muassa tehokkuus, hyödyllisyys, käytettävyys ja virheettömyys. Oikeastaan jälkimmäiset ovat saman tyyppisiä tekijöitä, joilla Nielsenin määritteli [32] tuotteen tai järjestelmän hyväksyttävyyden vuonna 1993.

Teollisuus otti käyttäjäkokemus-termin laajalti käyttöön 2000-luvulla. Akateemisen yhteisön käsitys termin merkityksestä kuitenkin poikkeaa teollisuuden käsityksestä. Akateemiset yhteisöt keskittyvät käyttäjäkokemuksen teoriaan, malleihin ja puitteisiin, joissa käsitellään nautinnollista näkökulmaa, tunteita, käytännön dynamiikkaa ja kanssakokemusta. Teollisessa tuotekehityksessä keskitytään käytännön käyttäjäkokemustyöhön tuotekehityksessä, jolloin fokuksessa ovat toiminnallisuudet, käytettävyys ja uutuustuotteen elinkaari. [41]

2.2 Käyttäjäkeskeiset suunnittelumenetelmät

Käyttäjäkeskeiset suunnittelumenetelmät projekteissa pitävät sisällään keinoja muun muassa tiedon hankkimiseen tuotteen käytöstä ja sen käyttäjistä, visualisoida ja organisoida projektin isoa kuvaa, visualisoida yksityiskohtaisia suunnitelmia ja luoda prototyyppejä. Käyttäjäkeskeisen suunnittelumenetelmän valinta voi olla suunnittelijoille hankalaa, eivätkä kaikki menetelmät täytä asetettuja vaatimuksia ja tarpeita. Tässä diplomityössä on jaettu menetelmät karkeasti käyttäjätiedon keräämiseen ja suunnitteluratkaisuja tukeviin menetelmiin, mutta todellisuudessa näiden välillä ei ole selkää linjaa, kuten Väänänen-Vainio-Mattila et al. artikkelissaan [41] toteavat. Artikkelissa huomautetaan, että teollisuudessa tarvitaan menetelmiä, jotka keskittyvät käyttäjäkokemukseen.

Käyttäjäkeskeinen suunnittelu koostuu eri vaiheista, kuten kuvassa 1 on kuvattu. Kuvan laatikot ”Ymmärrä ja määritä käyttökonteksti” ja ”Määrittele käyttäjän ja organisaation vaatimukset” kuuluvat esiselvitysvaiheeseen. ”Tuota suunitteluratkaisuja” ja ”Evaluoi suunnitteluratkaisut vaatimuksia vasten” kuuluvat suunnitteluvaiheeseen, jonka jälkeen tehdään toteutusta. Toteutuksen jälkeen tehdään testausvaihe. Vaiheita voidaan pilkkoa pienempiin vaiheisiin. Esimerkiksi, esiselvitysvaiheeseen saattaa projektista riippuen sisältyä konseptointivaihe, ja suunnitteluvaiheeseen visuaalisen suunnittelun vaihe.

2.2.1 Menetelmät käyttäjätiedon keräämiseen

Esiselvitysvaiheessa on tärkeää hankkia tietoa, jotta ymmärretään, mistä projektissa on kyse. Näin on helpompi ymmärtää kokonaiskuva ja tarvittavat resurssivaatimukset. Tiedon tarpeet ovat usein monitahoisia, eikä yhden menetelmän valinta tällöin riitä, vaan menetelmiä joudutaan yhdistelemään resurssien, tavoitteiden, käyttäjien ja suunnittelijoiden osaamisen puitteissa sopivalla tavalla. [19]

On huomioitava, ettei tiedonkeruu rajoitu vain projektin esiselvitysvaiheeseen, vaan käyttäjäpalautetta tulisi kerätä koko toteutusprosessin ajan. Eri menetelmin tehtyjä haastatteluja, kyselyitä, havainnointia ja työpajoja kannattaa järjestää hyvin varhaisessa vaiheessa suunnitteluprosessia. Alla olevassa taulukossa 1 on kootusti seuraavissa aliluvuissa esiteltävien tiedonkeruumenetelmien vahvuuksia, odotettavissa olevia tuloksia ja suurimpia rajoitteita.

Taulukko 1 *Yleisimpien tiedonkeruumenetelmien käyttökohteet, todennäköiset tulokset ja suurimmat riskit. Koottu lähteistä [19, 20, 34]*

	Käyttökohde	**Mahdolliset tulokset**	**Suurin rajoite**
Haastattelu	Käyttäjien toimien, tarpeiden, asenteiden ja mieltymysten yleispiirteiden selvittäminen.	Tietoa käyttäjien tavoitteista, syistä, prioriteeteista, arvoista ja mieltymyksistä. Tuloksena laadullista ja määrällistä tietoa riippuen menetelmästä.	Yksityiskohtien puute, tekemisen kaunistelu ja järkeistäminen. Vie paljon aikaa järjestää tilaisuus ja analysoida tuloksia.
Kysely	Vakiintunut terminoloa ja käyttäjäkunta, josta tiedetään paljon.	Tuotevertailu ja markkinointikartoitus. Pääasiallisesti määrällistä dataa.	Kaikki mistä kysyjät tai vastaajat eivät tiedä tuottaa arvailuja, kuten uusi teknologia, uudet käytöt ja käyttäjät. Kriittistä suunnitella ja valmistella

			hyvin. Vastausten saaminen usein rajallista.
Havainnointi	Yhteistyön, monimutkaisten käytäntöjen, rutiinien ja toissijaisen käytön selventäminen.	Käyttöympäristöä koskeva ymmärrys, vaihtoehtoiset suunnitteluideat, termit, käsitteet. Pääasiassa laadullista aineistoa.	Kohdennettava hyvin, ajallisesti hajautunutta, intiimiä toimintaa on vaikea tutkia. Menetelmän paljon aikaa vievä ja tuottaa paljon aineistoa, jonka analysointi on työlästä.
Työpaja	Innovatiivisten ratkaisujen löytäminen. Käyttäjien näkökulman mukaan ottaminen.	Luovat ideat, skenaariot, käyttäjien toiveet ja tarpeet. Tulokset laadullisia.	Ryhmädynamiikka ja jäsenten roolit voivat estää luovuuden. Osallistujien motivaation puute.

Jos asiakkaalla on jo olemassa tuote valmiina, ja tarkoituksena on parannella tai jatkokehittää sitä, suunnittelijat tekemät usein erilaisia asiantuntija-arvioita, jotka pohjautuvat erilaisiin heuristiikkoihin. Heuristiikkoja ja suunnitteluperiaatteita voidaan käyttää peruslinjausten ja lähtökohtien määrittelemiseen, mutta suunnittelussa tarvitaan myös konkreettista tuotekohtaista käyttäjätietoa. Asiakas ja käyttäjät maksavat viime kädessä kokonaisuuden, jonka tulee tuottaa heille hyötyä ja mielihyvää. Tuote on kokonaisuus, joka sisältää myös oheislaitteet, palvelut, verkot ja istuvuuden käyttäjien tavoitteisiin ja toimintatapoihin, joten kokonaisuuden hahmottamiseksi käyttäjätiedon kerääminen on keskeisessä osassa. [19]

2.2.1.1 Haastattelu

Haastattelulla selvitetään, mitä käyttäjä ajattelee, ja mitkä ovat hänen perimmäiset syynsä mielipiteille, toiveille ja toimintatavoille. Haastattelut ovat suhteellisen nopea ja joustava tapa kerätä käyttäjätietoa. Haastatteluilla saadaan koottua kuva käyttäjien toimien kokonaisuudesta, tavoitteista, haluista, arvomaailmasta, tehtävässä käytettävästä termistöstä, sekä työn ja teknologian ongelmakohdista. [19]

Haastattelut jaotellaan karkeasti strukturoituihin, puolistrukturoituihin ja avoimiin haastatteluihin. Strukturoitu haastattelu tarkoittaa kyselylomaketta, jossa kysymykset ja niihin tulevat vastausvaihtoehdot on määritelty etukäteen. Hyvässä lomakkeessa vastausvaihtoehdot ovat riittävän täydelliset ja pysyvät kohtuullisen pieninä. Tästä syystä sekä

lomakerakenne että kysymykset voivat olla jäykkiä. Lomake ei anna mahdollisuutta selittää syitä vastaukselle. Lomakkeen suunnittelijan pitää siis tarkkaan tietää, mitä ja miten hän asiaa kysyy. [94]

Puolistrukturoidussa haastattelussa (teemahaastattelu) osa kysymyksistä on strukturoituja ja osa avoimia. Haastateltavalle annetaan mahdollisuus selittää ja tukea vastauksiaan. Vapaalle tulkinnalle annetaan tilaa, vaikka haastatteluteemat on määritelty valmiiksi. Teemahaastattelu sopii tilanteisiin, joissa halutaan tietoa juuri tietystä aiheesta. Haastattelumuoto antaa mahdollisuuden uusien asioiden esiintuomiselle tunnetusta asiasta. [94, 88]

Strukturoimattomassa, eli avoimessa haastattelussa keskustelu ei ole formalisoitu. Avoin haastattelu etenee aihepiirin sisällä vapaasti haastateltavan ehdoilla. Haastattelussa on ennakkoon määriteltyjä teemoja, muttei niinkään tarkkoja kysymyksiä. Tarkoitus on haastatella keskustelumaisesti haastateltavan kokemuksista, muistoista, mielipiteistä ja perusteluista. Haastattelija on keskustelukumppani, joka ohjaa keskustelua aihepiirin ympärille ja kysyy syventäviä kysymyksiä. [88]

Haastatteluja voidaan tehdä yksilöille tai ryhmille (*engl. focus group*). Ryhmähaastattelussa osallistujia on yleensä kolmesta kymmeneen. Haastateltavat koostuvat eri kohderyhmien tai alojen edustajista. Tarkoituksena on saada osallistujat keskustelemaan tuotteesta, kehitysideoista, haasteista, tavoitteista ja odotuksista. Haasteena on saada kaikkien osallistujien ääni kuuluviin, ilma vahvempien persoonien dominointia. Haastattelutilannetta johtaa fasilitaattori, jonka tehtävä on antaa tilaa jokaiselle osallistujalle ilmaista mielipiteensä, ja pitää keskustelu aiheessa, sekä antaa erilaisia näkökulmia keskusteltavaksi. [19, 34]

Haasteet haastatteluissa tulevat materiaalin määrässä ja haastattelun tekemisen rationalisoinnissa. Haastatteluissa unohtuu helposti yksityiskohtien ja automatisoitujen suoritusten käsitteleminen. Haastattelu vaatii fokusointia ja valmistelua, jottei haastattelutilaisuus paisu. Materiaalin purkaminen ja analysointi vaativat aikaa ja ammattitaitoa. Haastattelutuloksia voi vääristää haastattelukysymysten vääränlainen muotoilu, kysymysten johdattelevaisuus ja haastattelijan tiedostamaton pyrkimys vaikuttaa käyttäjän ajatuksiin ja mielipiteisiin. [19]

2.2.1.2 Kyselyt

Kyselyissä saadaan kerättyä tietoa asiasta, jonka vastaajat ennalta tuntevat. Kyselyillä voidaan tavoittaa suuri määrä käyttäjiä, jolloin datan määrä kasvaa. Muihin tässä diplomityössä mainittuihin tutkimusmenetelmiin verrattuna kyselyiden toteuttaminen ja tulosten analysointi ovat kevyempiä. Strukturoituja vastauksia voidaan verrata toisiinsa ja muodostaa niistä tilastoja. Kyselyiden formaatit voivat vaihdella sähköposteista ja webkyselyistä perinteiseen postissa lähetettävään lomakkeeseen. [19, 32, 85]

Kyselyissä voidaan käyttää samoin kuin haastatteluissa strukturoituja, puolistrukturoituja ja avoimia lomakkeita. Avoimet kyselyt ovat erityisen vaativia sekä vastaajille että niiden analysoijille. Kyselyissä tutkija ei yleensä voi esittää tarkentavia kysymyksiä, jolloin kysymysten ja vastausvaihtoehtojen muodostaminen on kriittistä. Kyselyn pituus on tarkasti harkittava ja kysyttävä ainoastaan merkityksellisiä kysymyksiä, jotta vastaajan mielenkiinto säilyy, eikä vastaamista jätetä kesken. [32, 34]

2.2.1.3 Havainnointi

Havainnointi, eli observointi perustuu käyttäjän todellisen käyttöympäristön ja käyttäjien todellisen toiminnan ymmärtämiseen. Käyttäjiä seurataan heidän omassa käyttöympäristössään, jotta saadaan käsitys käyttäjän ja ympäristön yhteydestä ja vuorovaikutuksesta. Lisäksi menetelmällä selvitetään käyttäjien tehtäviä ja tarpeita. Havainnoinnin vahvuutena on tiedon kerääminen käyttäjän työstä, tavoitteista ja arvoista. Se tukee suunnittelukonseptin luomista. Menetelmän avulla voidaan löytää tuoteideoita tai havaita olemassa olevia puutteita. Parhaimmillaan voidaan selvittää tarkka työn kulku, monimutkaiset käytännöt, usean ihmisen käyttämät laitteistot ja toissijainen käyttö. [19, 34]

Havainnointia voidaan tehdä passiivisesti, jolloin tutkija tarkkailee käyttäjiä ympäristössään samalla tehden muistiinpanoja. Menetelmä sopii hyvin kokonaiskuvan hahmottamiseen. Varjostamishavainnoinnissa valitaan paikan sijasta henkilö, jota seurataan. Menetelmä soveltuu liikkuvan työn havainnointiin. Havainnointi haastattelussa (*engl. Contextual inquiy, CI*) käyttäjä työskentelee normaalisti, ja havainnoitsija seuraa ja esittää kysymyksiä, kun jokin toiminta tai syy toiminnan tekemiseen on epäselvä. Havainnoija voi myös pyytää käyttäjää kertomaan toiminnan ohella, mitä on tekemässä ja miksi. Tätä kutsutaan ääneen ajatteluksi (*engl. think aloud*). Havainnoinnin ongelmana on, ettei havainnoitsija tiedä, mitä käyttäjä ajattelee tehdessään valintoja ja päätöksiä, tai mitä ongelmia tämä kohtaa työnteon aikana. Käyttäjää häiritsee, jos havainnoitsija keskeyttää työskentelyn jatkuvilla kysymyksillä. [19, 30, 34]

Menetelmä on raskas ja aikaa vievä valmisteluineen ja analysointeineen. Menetelmä vaatii yleensä tunnusten luontia ja tutkimuslupien hankintaa. Tuloksista saatetaan tehdä yleistyksiä liian nopeasti, vaikka yleensä resursseja ei ole suurten joukkojen havainnointiin. Otannan sopivuus on siis tarkastettava. Havainnoijan on varottava, ettei keskity liiaksi hypoteesien oikeaksi todistamiseen. Menetelmässä pitäisi keskittyä asioihin, joita ei vielä tiedetä, eikä siihen, mitä tiedetään. [19, 32]

2.2.1.4 Työpaja

Työpaja on ryhmälähtöistä työskentelyä, jossa toimintaa helpotetaan ryhmän luovuudella ja ryhmän johtamisella yhteiseen päämäärään. Ryhmää johtava fasilitaattori on työpajan puolueeton osapuoli, joka keskittyy ryhmässä ilmeneviin rooleihin, tukee ryhmän luovuutta ja tarjoaa vaihtoehtoja ratkaisuille. [38] Työpajaan osallistuva ryhmä kokoon-

tuu yhteen. Kevyesti muodostettu työryhmä on helppo asetelma luovuuden ja spontaaniuden kannalta, koska sen jäsenillä ei ole määriteltyjä tehtäviä tai tavoitteita, toisin kuin yleensä tiimin jäsenillä. [18]

Ryhmädynamiikalla tarkoitetaan ryhmän sisäisten suhteiden, sen tuottavissa prosesseissa ja vuorovaikutuksessa tapahtuvia käänteitä. Näillä on suuri merkitys työpajan onnistumisen kannalta, sillä ne vaikuttavat ryhmän keskustelun avoimuuteen ja sujuvuuteen, ja ryhmän jäsenten välisiin tunteisiin. [20] Ryhmään muodostuu rooleja, kuten johtajia, kuuntelijoita, sivustaseuraajia, järjestelijöitä ja ongelmanratkaisijoita. [4] Ryhmätyöskentelyn haasteita ovat muun muassa kaikkien jäsenten osallistuminen. Äärimmäisyyksissään tilanne voi kehkeytyä jäsenten väliseksi valtataisteluksi, joka rikkoo ryhmän avoimuuden ja luovuuden. Fasilitaattorin tehtävä on mahdollistaa tasapuolinen osallistuminen, ja näin luovuuden lisääminen [9, 20, 22]

2.2.2 Menetelmät suunnitteluratkaisujen ja kehityksen tueksi

2.2.2.1 Persoonat, käyttäjätarinat ja käyttäjäeepokset

Persoonia käytetään käyttäjäkeskeisen suunnittelun ja markkinoinnin työkaluna. Ne ovat hypoteettisia kuvauksia erilaisista sovelluksen potentiaalisista käyttäjistä. Niissä kuvataan käyttäytymismalleja, tavoitteita, taitoja, asenteita ja ympäristöä. Persoonat ovat hyvä keino hahmottaa käyttäjän tavoitteita, toiveita ja rajoitteita, jotka vaikuttavat päätöksentekoon järjestelmän toiminnallisuuksista, interaktiosta ja visuaalisesta suunnittelusta. Persoonat perustuvat kerättyyn käyttäjätietoon. [27, 77]

Persoonista on hyötyä koko projektitiimille, kun niiden kautta voidaan ilmaista potentiaalisten käyttäjien variaatiota eri käyttökonteksteissa. Koko tiimi voi peilata ja priorisoida suunniteltuja ratkaisuja persoonien osaamisiin ja rajoitteisiin. [10]

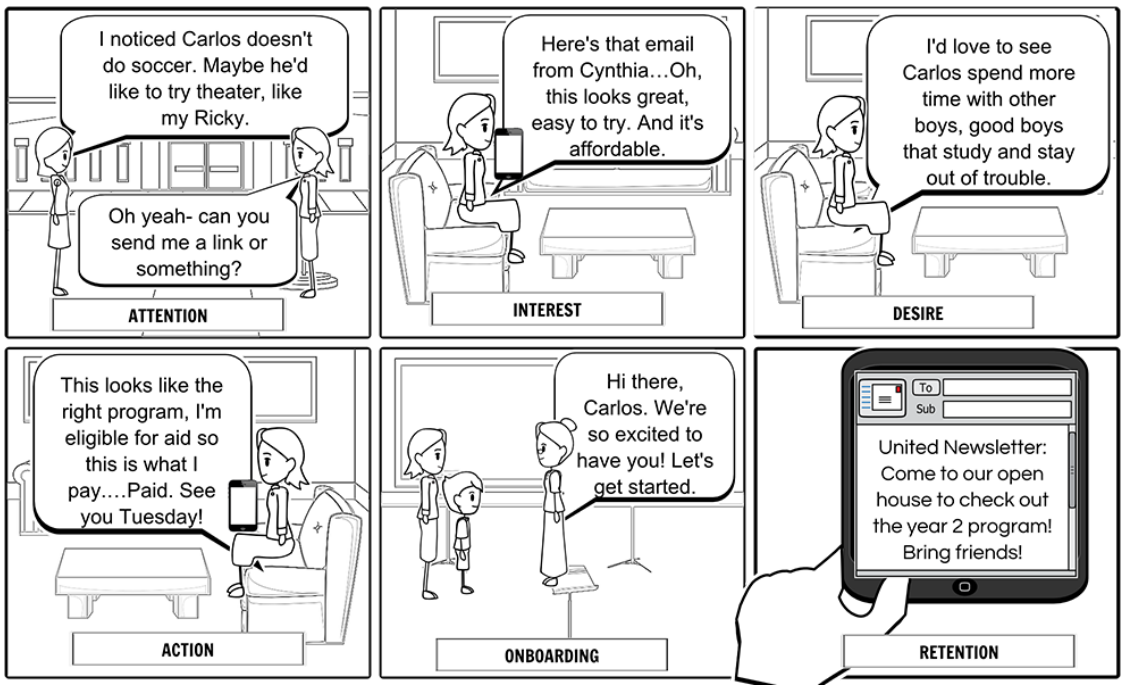
Käyttäjätarinalla (*engl. user story*) kuvataan järjestelmän toiminnallisia vaatimuksia korkealla tasolla ja ymmärrettävässä, tarinankaltaisessa muodossa. Käyttäjätarinalla kuvataan lyhyesti ’*kuka* tekee’ ja ’*mitä* tekee’, sekä ’*miksi* tekee’, eli se antaa juuri ja juuri tarpeeksi tietoa, jotta kehittäjät pystyvät tekemään järkevän työmääräarvion. Pohjana käyttäjätarinoille voidaan käyttää persoonista muodostettavia käyttäjäryhmiä, toimijoita (*engl. actor*). Järjestelmällä on useampia käyttäjätarinoita, joista voidaan hahmottaa kokonaiskuva järjestelmästä myös kehittäjille. [68, 80] Käyttäjätarina voi olla esimerkiksi seuraavienlainen:

- Opiskelijat voivat ilmoittautua kursseille portaalin kurssi-osion kautta.
- Opiskelijat näkevät kurssin aikataulun portaalin kurssi-osion kautta.
- Opiskelijat voivat ilmoittautua kurssin tentteihin portaalin kurssi-osion kautta.
- Opiskelijat voivat perua kurssi-ilmoittautumisen portaalin kurssi-osion kautta.

Ketterien ohjelmistokehitysmenetelmien yhteydessä suunnittelijat käyttävät käyttäjätarinoista toisinaan termiä käyttäjäeepos (*engl. epic*), jolla viitataan olennaisesti normaalia suurempaan käyttäjätarinakokonaisuuteen. Tyypillisesti käyttäjäeepokset ovat alhaisen prioriteetin käyttäjätarinoita, joita ei voida pilkkoa yksiselitteisesti pienempiin suunnittelukokonaisuuksiin, eikä sitä voi toteuttaa vain yhden sprintin aikana. Käyttäjäeepos voidaan, ja se pitää pilkkoa kuitenkin toteutettaviin kokonaisuuksiin, jotta sitä voidaan hallita ja edistää prosessissa. Käyttäjäeepoksen toteutuksen lopuksi on varmistuttava, että alkuperäisen käyttäjäeepoksen vaatimukset toteutuvat. [7, 68]

2.2.2.2 Skenaariot ja storyboard

Skenaario on tarina, jolla kuvataan tuotteen käyttömahdollisuutta tehtävän saavuttamiseksi. Storyboard on kuvakäsikirjoitus, jossa tapahtumat ja tehtävät visualisoidaan sarjakuvamaisesti. Storyboard voidaan siis muodostaa skenaarioista jakamalla skenaariot askeliin, joissa käyttäjä on vuorovaikutuksessa sovelluksen kanssa, ja luo kokonaisen tapahtuman storyboardille. Molempia työkaluja voidaan käyttää suunnittelun evaluoinnin kohteena, ja varmistamaan, että koko konteksti ja tarpeet on otettu huomioon. [34] Alla olevassa kuvassa 2 on esimerkki storyboardista.



***Kuva 2** Esimerkki storyboardista [70]*

Storyboardin tarkoitus on helpottaa tuotantoprosessia kuvaamalla käyttöliittymiä tai kertoa järjestelmän toiminnallisuus sarjakuvatarinana, ja elävöittää kokonaiskuva, visio ja konsepti sekä asiakkaille että projektitiimille. Storyboardiin kiteytyy persoonat, käyttäjätarina, vaatimukset, erilaiset rajoitteet ja ratkaisut. Sen avulla voidaan kerätä pa-

lautetta käyttäjiltä ja projektitiimin muilta henkilöiltä, sekä pyytää suunnittelutiimiä tarkentamaan järjestelmän käyttöä yksityiskohtaisemmin. Lisäksi storyboardia voidaan käyttää potentiaalisten skenaarioiden matalantason testaamiseen. [34, 71]

2.2.2.3 Rautalankamallit, prototyypit ja käyttäjätestaus

Rautalankamallit (*engl wireframes*) ovat kevyin mahdollinen sovelluksen määrittelydokumentti, joka tuotetaan suunnitteluprosessin aikana. Rautalangat voidaan piirtää kynällä paperille, tussilla taululle tai erilaisilla työkaluilla, kuten Balsamiq [47] ja Sketch [62]. Rautalankamallit eivät ole viimeisteltyjä versioita käyttöliittymästä, vaan luonnoksia, joita voidaan helposti muokata. Niissä keskitytään saatavilla oleviin toiminnallisuuksiin, tietojen ja toimintojen keskinäiseen tärkeysjärjestykseen, tiedon esittämiseen liittyviin sääntöihin ja erilaisten skenaarioiden vaikutuksiin näytöllä. [7, 8]

Rautalankamalleja ja prototyyppejä rakennetaan iteratiivisesti käyttäjäkeskeisen suunnitteluprosessin tavoin. Ne ovat nopeita, halpoja ja toimivia keinoja ratkaista ongelmia ja varmistua suunnitteluratkaisuista, muun muassa sovelluksen keskeisten toimintojen osalta. Prototyyppi paljastaa jo suhteellisen varhaisessa vaiheessa projektia tuotteiden puutteita ja ongelmia konkreettisella tasolla. Mobiili- ja web-applikaatioiden prototyyppi voidaan rakentaa jo todelliseen käyttöympäristöön staattiseksi tai puolistaattiseksi erilaisten työkalujen avulla. [19] Tällaisia työvälineitä ovat esimerkiksi InVision [53] ja Framer [51].

Prototyyppien rakentaminen ja esittäminen ovat hyvä tapa kerätä käyttäjäpalautetta tuotteesta tai ideasta, joka on valmis esitettäväksi. Prototyypin avulla voidaan käyttäjille ja asiakkaille esitellä malli, miten suunnittelijat ovat ymmärtäneet tarpeet ja tavoitteet. Käyttäjät voivat joko vahvistaa tai korjata oletuksia. Prototyyppien avulla voidaan selvittää muun muassa käyttöliittymän helppokäyttöisyyttä ja intuitiivisuutta, tai tuotteen järkevää rakennetta. Prototyyppivaiheessa käyttäjiltä saatava palaute on suunnittelijoille ja kehittäjille arvokasta, kun tuotteen lopullinen kehittäminen ei ole vielä välttämättä alkanut, ja muutokset ovat vielä suhteellisen helppoja ja halpoja. Ongelmien havaitseminen ja korjaaminen projektin myöhäisessä vaiheessa aiheuttaa projektille kertautuvia resurssivaatimuksia ja kustannuksia. [19]

Käyttäjätestausta voidaan tehdä rautalankamallien tai prototyyppien pohjalta. Käyttäjätestauksella pyritään saamaan vahvistus sille, että tuotteen kehitystä voidaan viedä eteenpäin ja tuotteen käytettävyys on hyvä. Käyttäjätestausta varten suunnittelijat valmistelevat tehtäviä käyttäjille, ja observoivat ja mittaavat tehtävien suorittamista. Testaukseen voi liittyä myös kyselyitä, haastatteluja ja ryhmäkeskusteluja eri menetelmin. Tuloksista saadaan aikaan muutosehdotuksia, jotka priorisoidaan ja otetaan mukaan toteutukseen, jos ne koetaan tarpeeksi kriittisiksi. [21, 34]

3. KETTERÄT MENETELMÄT

Yritykset ovat siirtyneet perinteisestä vesiputousmallista kohti ketteryyttä, jonka on todettu palvelevan projektin eri osapuolia paremmin kuin vesiputousmalli. Ketterissä ohjelmistoprojekteissa otetaan huomion projektissa toimivan kehitystiimin ja eri sidosryhmien yhteistyö muun muassa läpinäkyvyyden avulla. Läpinäkyvyys sidosryhmien ja kehitystiimin välillä parantaa kommunikaatiota osapuolten välillä huomattavasti, ja näin ketteryys mahdollistaa tuotteen kehityksen, jolla on liiketoiminnallista arvoa ja paras mahdollinen laatu. Liiketoiminnallinen onnistuminen ei ole todennäköistä ilman keskittymistä loppukäyttäjien tarpeisiin, johon ketteryydessäkin keskitytään. Tästä syystä käyttäjäkeskeisellä suunnittelulla on tärkeä rooli ketterässä ohjelmistoprojektissa. Eri sidosryhmien ja loppukäyttäjien tarpeiden tarkentuessa ajan myötä lisääntyy myös ymmärrys lopputuotteen vaatimuksista, jolloin ketterien menetelmien periaate muutosten mahdollistamisesta osoittaa paremmuutensa verrattuna perinteiseen vesiputousmalliin. Muutokset projektin aikana ovat mahdollisia, koska lopputuotetta ei suunnitella projektin alussa valmiiksi, vaan alussa luodaan visio ja iso kuva, mihin ollaan tähtäämässä.

Vesiputousmalli edelsi ketteriä menetelmiä, ja loi pohjaa ohjelmistoprojekteissa toteutettaville vaiheille, jotka ovat ketteryydessäkin tärkeitä, vaikka niitä toteutetaan eri tavalla kuin perinteisessä vesiputousmallissa. Vesiputousmallisesta kehitysprosessista voidaan erottaa selkeästi neljä vaihetta: määrittely, suunnittelu, ohjelmointi ja testaus. Lisäksi näiden jälkeen seuraa ohjelmiston käyttöönotto ja ylläpito, jotka ovat osa tuotteen elinkaarta. Tuotteen elinkaarella tarkoitetaan aikaa tuotteen kehitysvaiheesta sen poistamiseen käytöstä. Edellä kuvattua vaihejakoa kutsutaan vesiputousmalliksi, josta on olemassa useita erilaisia muunnelmia. [17] Alkuperäisen vesiputousmallin määritteli Winston W. Royce artikkelissaan *Managing the Develoment of Large Software Systems* [35].

Royce pyrki korostamaan mallissa iterointia taaksepäin, ja huomautti, että tuotteen ongelmat huomataan yleensä vasta sen testausvaiheessa, jolloin kustannusarvio voi helposti kaksinkertaistua. Teollisuudessa vesiputousmalli kuitenkin yksinkertaistettiin, ja taaksepäin kulkevat iteraatiot unohdettiin, ja kehitysprosessin vaiheet eroteltiin täysin toisistaan, jolloin seuraavaa vaihetta ei aloitettu ennen kuin edellinen oli kokonaan valmis, koska edeltävä vaihe loi edellytykset seuraavalle vaiheelle. Vaiheiden evaluointi vaikutti suoraan kaikkiin kohteen jälkeisiin vaiheisiin, josta muodostui mallin haaste: mitä pidemmälle kehitystyö vietiin, sitä vaikeampaa ja kalliimpaa muutosten tekeminen

oli. Royce painotti asiakaskontaktin tärkeyttä, ja kehotti useisiin tuotteen läpikäynteihin asiakkaan kanssa, jotta lopputuote olisi tyydyttävä. [17, 19, 35]

Ajan kuluessa yrityksen havahtuivat siihen, että muutosten mukaanotto vesiputousmalliin ei toiminut, eikä ongelmiin reagoitu ajoissa. Kommunikaatio oli heikkoa ja nojasi dokumentaatioihin, joka ei lisännyt tuotteelle arvoa, vaikka se vei paljon resursseja. 2000-luvun taitteessa haasteet ajoivat ohjelmistoyritykset uusimaan kehitysprosessejaan nopeimmiksi ja joustavimmiksi. [7, 19]

Vuonna 2001 joukko ohjelmistotuotannon asiantuntijoita määritteli ketterien menetelmien arvot. Tätä kokoelmaa kutsutaan *Ketterän ohjelmistokehityksen julistukseksi* (*engl. Agile Manifesto*). Heidän määrittelemänsä arvot ovat:

- Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja,
- Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota,
- Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja,
- Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa. [58, 64]

Arvojen lisäksi määriteltiin kaksitoista pääperiaatetta, joissa korostuu lisäarvon tuottaminen asiakkaalle mahdollisimman aikaisessa vaiheessa, toimivien versioiden toimittaminen säännöllisin aikavälein, muutosten mukaan ottaminen milloin vain, yhteistyö ja kommunikaatio sidosryhmien ja kehitystiimin välillä, sekä yksinkertaisuus ja tehokkuus. [58, 64]

Ketterän ohjelmistokehityksen julistuksen arvojen ja periaatteiden puitteissa on syntynyt useampia ketteriä menetelmiä, ja työkaluja tukemaan menetelmiä [7]. Seuraavissa kappaleissa käsitellään kahta Suomessa yleisimmin käytössä olevaa menetelmää, Scrumia ja Leania, sekä Kanban-taulua.

3.1 Scrum

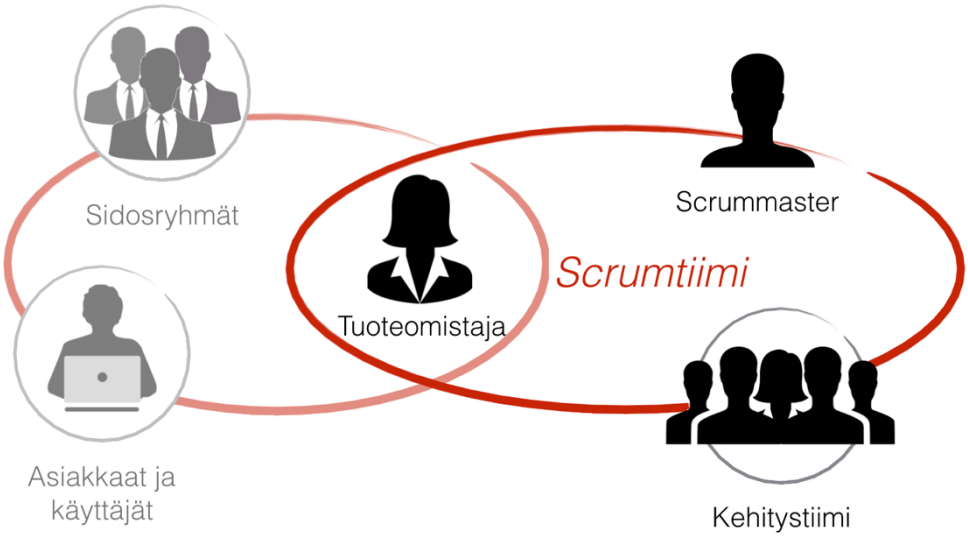
Scrum on viitekehys, jonka tarkoituksena on antaa tuotteelle lisäarvoa tuottavalla ja luovalla tavalla. Scrumin on kevyt ja sen konsepti on helppo ymmärtää. Scrumilla saadaan selkeyttä monimutkaisten tuotteiden kehittämisprosesseihin ja organisoitua kehitystiimin työtä. Toisaalta taas Scrumia on vaikea hallita hyvin. [89]

Scrum koostuu scrumtiimistä, jossa erilaiset roolit, tuotokset ja säännöt sitovat yhteen eri osa-alueet, ja ohjaavat tiimin vuorovaikutusta. Vaikka viitekehykseen kuuluu monia osia, on tarkoitus ottaa niistä käyttöön vain ne, jotka palvelevat kyseistä projektia ja tiimiä. On siis tarkoituksenmukaista, että toimintatavat vaihtelevat projekteittain. [89]

Scrumissa iteratiivisuudella voidaan optimoida ennustettavuutta ja kontrolloida riskejä. Sen kolme periaatetta ovat läpinäkyvyys, tarkastelu ja sopeuttaminen. Läpinäkyvyys on muun muassa tärkeiden tekijöiden ennalta määrittelyä ja tuotteen hyväksyjien yhteistä määritelmää tuotteen tai osakokonaisuuden valmis-tilalle. Tarkastelulla viitataan Scrumin käyttäjien säännölliseen tuotosten tarkasteluun ja työn edistymisen seurantaan, jolla pyritään havaitsemaan ja reagoimaan poikkeamiin ajoissa ja nopeasti. Sopeuttaminen merkitsee prosessin ja materiaalien säätämistä, kun poikkeama on havaittua. [89]

3.1.1 Roolit ja tehtävät

Scrumtiimissä on kolme erilaista roolia ja jokaisella on omat määritellyt tehtävät. Roolit ovat tuoteomistaja (*engl. product owner*), scrummaster ja kehitystiimi. [7] Alla olevassa kuvassa 3 on havainnollistettu Scrumtiimin kokoonpanoa, ja heidän suhdetta toisiinsa.



Kuva 3 *Scrumtiimin keskinäiset sidokset. Muokattu lähteestä [44]*

Kehitystiimin on kooltaan noin viidestä yhdeksään henkilöä, lukuun ottamatta scrummasteria tai tuoteomistajaa. Kehitystiimin koko vaihtelee riippuen projektin koosta, aikataulusta, resursseista ja vaatimuksista. Siinä ei ole perinteisiä ohjelmistokehitysprosessien rooleja, kuten arkkitehtiä, ohjelmoijaa, käytettävyyssuunnittelijaa ja testaajaa, vaan se on itseohjautuva ryhmä, jonka jäsenet päättävät keskenään kuka tekee mitäkin. Kehitystiimi määrittelee yhdessä, milloin osakokonaisuudet valmiita. Kehitystiimi antaa osakokonaisuuksille työmääräarviot, ja jakaa ne pienemmiksi tehtäviksi (*engl. tasks*), joiden koot voivat vaihdella muutamasta tunnista pariin päivään. [12, 89]

Scrummaster on scrumtiimin jäsen, joka mahdollistaa kehitystiimin toiminnan esteettömyyden, muttei tuo kehitystiimiin teknistä panosta tai pyri vaikuttamaan sen päätöksen-

tekoon. Scrummaster lisäksi organisoi ja johtaa kokouksia ja avustaa tuoteomistajaa muun muassa fasilitoimalla scrum-tapahtumia, sekä johtamalla ja valmentamalla organisaatiota Scrumin käyttöönotossa [12, 89]

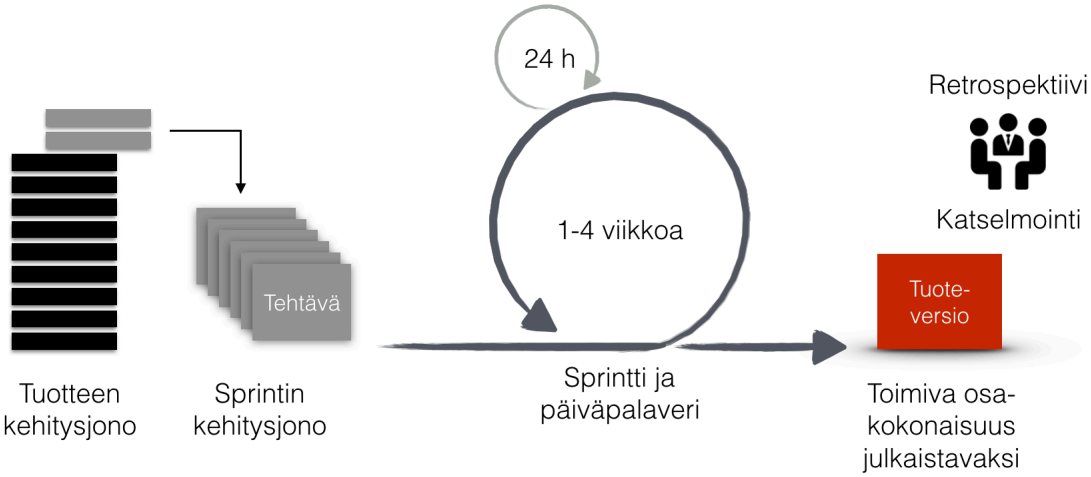
Tuoteomistaja edustaa projektissa sidosryhmiä ja liiketoimintaa optimoimalla sijoitetun pääoman (*engl. Return of Investment, ROI*) tuotto yritykselle. ROI voi tarkoittaa muun muassa asiakastyytyväisyyttä, rahallisia etuja tai ympäristöarvoja. Tuoteomistaja toimii kontaktina scrumtiimin ja asiakkaan välillä olematta asiakkaan edustaja. Hänen tärkein tehtävänsä on priorisoida ja päivittää tuotteen kehitysjonoa, ja varmistaa kehitysjonon avoimuus, läpinäkyvyys ja ymmärrettävyys sekä scrumtiimille, että asiakkaalle ja sidosryhmille. Tuoteomistaja päättää, milloin sprintti, josta kerrotaan seuraavassa kappaleessa lisää, on valmis toteutettavaksi. [12, 89]

3.1.2 Prosessi ja tapahtumat

Scrum ei ole prosessinhallintamenetelmä, vaikka se perustuukin empiiriseen prosessinhallintaan. Scrumissa on ennalta sovittuja tapahtumia, jotka luovat säännöllisyyttä ja minimoivat muut kuin itse Scrumiin liittyvät palaverit. Kaikki Scrumiin liittyvät tapaamiset ovat aikarajattuja, ja ne lisäävät läpinäkyvyyttä ja mahdollistavat sopeuttamisen. [89]

Alussa asiakkaan ja käyttäjien vaatimusten, tarpeiden, toiveiden ja käyttäjätarinoiden pohjalta luodaan tuotteen kehitysjono (*engl. Product Backlog, PBL*). Tuotteen kehitysjono on lista kaikesta, mitä tuotteeseen tarvitaan, ja ainoa lähde myöhemminkin tuotteeseen tuotaville muutoksille ja vaatimuksille. Tuotteen kehitysjono ei ole siis koskaan valmis dokumentti, vaan se elää projektin edetessä ja tietämyksen lisääntyessä. Tuoteomistajan tehtävä on huolehtia tuotteen kehitysjonosta. [80, 89]

Ennen sprintin aloittamista pidetään sprintin suunnittelupalaveri, jossa päätetään sprintin sisältämät työt, selvitetään, mitä sen aikana on mahdollista toteuttaa, ja miten työ tulisi jakaa ja toteuttaa kehitystiimin kesken. Tehtävät on kuvattu käyttäjätarinoina ja työmäärän arviointi tehdään koskien koko käyttäjätarinaa. [14, 89] Tuotoksena palaverista syntyy sprintin kehitysjono. (*engl. Sprint Backlog, SBL*), josta kehitystiimin jäsenet ottavat itselleen kohtia työn alle ja jakavat ne pienempiin kokonaisuuksiin, tehtäviin. Palaverin kesto on enimmillään kahdeksan tuntia. Kehitystiimi myös suunnittelee sprintin työnkulun. Tuotoksena palaverista syntyy sprintin kehitysjono. [89] Alla olevassa kuvassa 4 on havainnollistettu Scrumin mukaista prosessimallia vaiheineen ja tapahtumineen.



Kuva 4 *Scrum-prosessi. Muokattu lähteestä [75]*

Sprintti (*engl. sprint*) on koko prosessin ydin. Sen aikana tuotetaan ”valmiin” määritelmän täyttävä, käyttökelpoinen ja potentiaalisesti julkaisukelpoinen tuoteversio. Sprintti kestää enimmillään neljä viikkoa, jolla varmistetaan jatkuva tuoteversioiden toimittaminen asiakkaalle. Sprintit suoritetaan toinen toistensa perään. Sprintin aikana ei tulisi tehdä vaatimuksiin tai määritelmiin muutoksia, jotka vaikuttavat meneillä olevan sprintin tavoitteisiin. Niitä voidaan kuitenkin tarkentaa ja ratkaisuista voidaan neuvotella scrumtiimin sisällä. Tarvittavat muutokset tehdään tulevien sprinttien aikana, kun ne on määritelty ja priorisoitu uudelleen. Sprintti voidaan keskeyttää, jos sen tavoite ja sisältö muuttuvat oleellisesti tai siitä tulee tarpeeton. [89]

Scrumtiimi pitää päivittäin päiväpalaverin (*engl. daily scrum*), joka kestää 15 minuuttia. Päiväpalaverissa scrumtiimi käy läpi kaikkien jäsenten työtilanteen, keskeiset ongelmat ja luo suunnitelman seuraavalle 24 tunnille. Päiväpalaveri pyritään pitämään seisten, jotta varmistuttaisiin, ettei palaveri veny tarpeettomasti. [89]

Kun sprintti on päättynyt, kaikki sprintin kehitysjonolle otetut tehtävät tulisi olla ”valmis” –tilassa. Tulokset käydään läpi sprintin katselmoinnissa. Tilaisuudessa käydään läpi kehitetty versio ja tarvittaessa sopeutetaan tuotteen kehitysjonoa, jos havaitaan muutostarpeita. Katselmointi pidetään yhdessä sidosryhmien edustajien ja scrumtiimin kanssa. Tavoitteena on kerätä palautetta, edistää osapuolten välistä kommunikointia ja ennakoida seuraavan sprintin suunnittelupalaveria. [89]

Sprintin retrospektiivi on scrumtiimin oma palaveri, jossa se voi käydä läpi omaa suoriutumistaan ja työn sujuvuutta, sekä ehdottaa parannuksia kehitysprosessiin. Retrospektiivi ajoitetaan katselmoinnin ja seuraavan sprintin suunnittelupalaverin väliin.

Scrummasterin on velvollisuus varmistaa, että retrospektiivi pidetään, ja ohjaa käsiteltävien asioiden läpikäynnissä. [89]

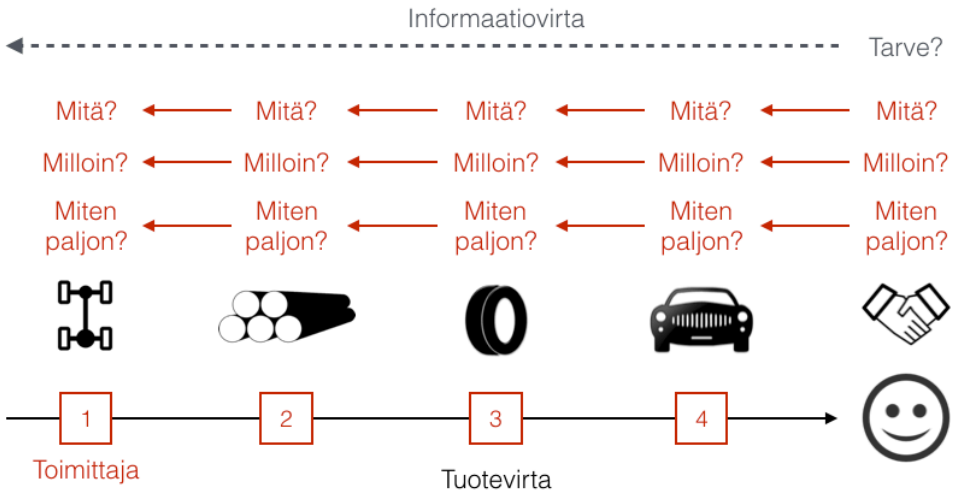
3.2 Lean

Lean on kehitetty alun perin Japanissa Toyotan tuotantoprosessiperiaatteiden pohjalta. Toyota valmisti autoja kotimaansa markkinoille toisen maailman sodan jälkeen, jolloin teollisuutta rajoitti resurssipula. Toyotan sisäistä tuotantofilosofiaa, jota Taiichi Ohno oli kehittämässä ja kirjoitti aiheesta kirjan vuonna 1978 [33], kutsutaan Toyota Production System:ksi (TPS). Myöhemmin länsimaiset tarkkailijat nimesivät virtaustehokkaan tuotantoprosessin Leaniksi. Vaikka Leanin käsite on luotu Toyotan lähtökohdista, se on kuitenkin eri käsite kuin TPS. [24, 31]

3.2.1 Leanin lähtökohdat

Tuotantokehityksen ydin oli tehdä oikeita asioita ja tehdä asiat oikein. Oikeiden asioiden tekeminen korostui sodanjälkeisenä aikana, jolloin virheinvestointeihin ei ollut varaa, ja asiakkaat ostivat vain sitä mitä tarvitsivat ja halusivat. Asiakastarpeiden kartoituksesta tuli myös tärkeä osa tuotantoa. Toyota ymmärsi panostaa vuorovaikutukseen asiakkaan kanssa, jolloin yrityksen oli helpompi kartoittaa toiveita ja tarpeita, ja muokata tuotantoa vastaamaan kysyntään. [31]

Käyttöön otettiin tilauslähtöinen tuotanto, imuohjausjärjestelmä, jossa tuotanto alkaa vasta tilauksen saavuttua. Tilaus kulki asiakkaalta vastavirtaan tuotantoprosessin läpi, jolloin koko prosessi sai tiedoksi, mitä piti tuottaa, milloin ja kuinka paljon. Tuotantoprosessi nähtiin yhtenä virtauksena, vaikka se koostuin eri tuotantovaiheista. Tätä prosessia on kuvattu alla olevassa kuvassa 5. Jokaisella vaiheella oli kaksi rooli: toimittaja edeltäjälleen, ja asiakas jälkeiselleen. Alla olevassa kuvassa on havainnollistettu virtauksen jakautumista neljään vaiheeseen. Neljäs vaihe ottaa loppuasiakkaalta tilauksen vastaan, ja toimii näin tälle toimittajana. Tämän jälkeen vaihe neljä toimii asiakkaana vaiheelle kolme, ja esittää tälle omat vaatimuksensa. Seuraavissa vaiheissa malli toistuu. Näin saadaan tuotettua asiakkaalle ja tuotteelle arvoa virtauksen jokaisessa vaiheessa. [31]



Kuva 5 *Virtaus tuotantoprosessissa [31]*

Imuohjausjärjestelmä tuki myös asioiden oikein tekemistä, jonka tarkoitus oli välttää sitouttamasta pääomaa varastoihin ja välivarastoihin. Sisäisen jakelun tuli toimia moitteettomasti, jotta virtaus olisi mahdollisimman nopea raaka-aineista valmistukseen, toimittamiseen ja maksun saamiseen. Kaikki hukan muodot, jotka eivät tuottaneet lopputuotteelle arvoa eliminoitiin virtaustehokkuuden parantamiseksi. Tällaisia olivat muun muassa turha odottelu, liikatuotanto, tarpeeton tuotanto, tarpeeton varastointi ja tarpeettomat virheet, työn tekeminen uudelleen tai päällekkäin. [31]

Asioiden oikein tekeminen tarkoitti myös tuotteiden virheettömyyttä, joten tuotannonohjaus ja laadunvalvonta olivat tärkeissä rooleissa. Kohdattuihin ongelmiin suhtauduttiin rakentavasti ja positiivisesta. Ne nähtiin asioina, jotka piti tunnistaa, analysoida ja eliminoida. Virheitä ei saanut päästää asiakkaille asti. [31]

Vuonna 1988 John Krafcikin lanseerasi artikkelissa *Lean-tuotantojärjestelmän riemuvoitto* ensi kertaa käsitteen *'Lean production'*. Artikkelissa argumentoitiin myös tuottavuuden ja laadun lisääntyvän Lean-tuotantojärjestelmän avulla, eikä niinkään mittakaavaedulla ja huipputekniikalla. James P. Womack, Daniel T. Jones ja D. Roos [43] argumentoivat 1990 Leanin koostuvan neljästä periaatteesta. Vuonna 1994 Womack ja Jones kokosivat viisi Leanin toteutukseen painottuvaa periaatetta teokseen *Lean Thinking* [42]. Yksi näistä oli parannusten jatkuva iterointi, kunnes kaikki hukka on eliminoitu. Kehitetyt periaatteet ovat vaikuttaneet Lean-käsitteen kehittymiseen ja leviämiseen enemmän kuin mikään muu kirjallinen lähde. Vuonna 2001 Toyota laati sisäisen kirjoituksen ”*Toyota Way*”, jonka tavoitteena oli yhdenmukaistaa Lean-näkemys koko globaalissa yrityksessä viiden arvon avulla, jotka oli jaettu kahteen kokonaisuuteen, jatkuva parantaminen ja kunnioitus ihmisiä kohtaan. Leanin perusteokseksi palvelualoil-

la on tullut Jeff K. Likerin 2004 julkaisema *The Toyota Way* [28], jossa kirjailija tulkitsee Toyotan filosofian paketoimalla sen neljääntoista pariaatteeseen.

I Pitkäjänteinen filosofia
    1. Pohjaa päätökset pitkäjänteiseen filosofiaan, vaikka se tapahtuisi lyhytaikaisten taloudellisten tavoitteiden kustannuksella

II Oikea prosessi tuottaa oikean tuloksen
    1. Luo jatkuva virtaus, jotta ongelmat tulevat esille.
    2. Anna kysynnän ohjata, jotta vältytään liikatuotannolta.
    3. Tasaa työkuorma.
    4. Pysäytä tarvittaessa prosessi ongelmien ratkaisua varten, jotta laatu on alusta pitäen oikea.
    5. Vakioitu työ on perusta jatkuville parannuksille ja henkilöstön osallistumiselle.
    6. Käytä visuaalista ohjausta, jotta ongelmat eivät jää piiloon.
    7. Käytä vain luotettavaa, hyväksi havaittua tekniikkaa, joka sopii henkilöstölle ja prosesseille.

III Huolehdi työntekijöiden ja kumppaneiden kehittämisestä
    1. Kouli johtajia, jotka todella ymmärtävät työtä, jotka noudattavat filosofiaa ja opettavat sitä muille.
    2. Huolehdi yrityksen filosofiaa noudattavien poikkeuksellisten ihmisten ja tiimien kehittämisestä.
    3. Kunnioita kumppaneita ja toimittajia heittämällä heille haasteita ja auttamalla heitä kehittymään.

IV Jatkuva työskentely toiminnan perusongelmien kanssa edistää organisaation oppimista
    1. Käy katsomassa paikan päällä, jotta ymmärrät tilanteen kunnolla
    2. Tehkää päätöksiä hitaasti ja yhteisymmärryksessä. Toteuttakaa päätökset nopeasti.
    3. Kehittykää oppivaksi organisaatioksi väsymättömän pohtimisen ja jatkuvien parannusten kautta. [31]

Ajan myötä eri tahot ovat kehittäneet Leanista useita konsepteja soveltaen sen arvoja ja periaatteita omille toimialoilleen, myyntiin, terveydenhuoltoon ja IT-alalle. Samalla Leanin ympärille on muodostunut laaja-alaista käsitteistöä, kuten Lean suunnittelu, Lean johtaminen ja Lean muutos. Leanin laajentuminen uusille toimialoille ja uusiksi käsitteiksi vaikeuttaa Leanin täsmällistä määrittelemistä. [31]

3.2.2 Leanin peruskäsitteet

Leaniin ja sen ymmärtämiseen liittyy tiettyjä haasteitta, käsitteitä ja periaatteita, joita tässä kappaleessa käydään läpi lyhyesti, jotta Leanin itsensä sekä myöhemmin esiteltävien tutkimustulosten ymmärtäminen olisi helpompaa.

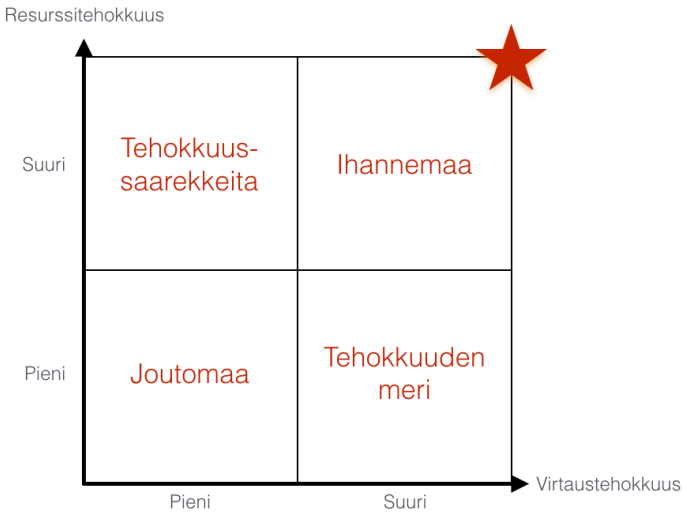
3.2.2.1 Resurssi- ja virtaustehokkuus

Resurssitehokkuus on organisaatioissa perinteinen mittari, jossa korostuu arvoa tuottavien resurssien tehokas hyödyntäminen, eli kapasiteetin maksimaalinen käyttö ja rahalle saatavan vastineen maksimointi. Resurssitehokkuus mittaa resurssien hyödyntämistä suhteessa tiettyyn ajanjaksoon. Resurssitehokkuuden periaatteena on pilkkoa tehtävät pieniin osiin, ja osoittaa niitä eri ihmisille, funktioille ja organisaatioille tehtäväksi. Toinen periaate on niputtaa pieniä tehtäviä yhteen, jolloin eri yksiköt saattavat tehdä päällekkäistä työtä. Hyvän resurssitehokkuuden takaamiseksi on tärkeää pitää resurssit jatkuvassa käytössä varmistamalla, että näillä on jatkuvasti jokin virtausyksikkö jalostettavana. Liiallinen resurssitehokkuuteen panostaminen johtaa siihen, ettei resursseilla ole kapasiteettia ottaa uusia projekteja ja tehtäviä vastaan, ja käytännössä resursseille osoitetaan ylikuormaa. Resurssitehokkuuteen keskittyminen tuottaa virtausyksiköille paljon toissijaisia tarpeita, jotka osaltaan täyttävät ensisijaisilta tehtäviltä ylijääneen ajan. Resurssitehokkuuden maksimointi tehdään virtaustehokkuuden kustannuksella. [31]

Virtaustehokkuudessa päähuomio on jalostettavassa yksikössä, kuten tuotteessa tai potilaassa. Siinä avainasemassa on aika, joka kuluu asiakkaan tarpeen tunnistamisesta sen tyydyttämiseen. Virtaustehokkuus syntyy (tuotanto)prosesseissa, joista pyritään poistamaan kaikki jalostettavalle yksikölle arvoa tuottamattomat toiminnot, eli kaikki hukka. Prosessilla tarkoitetaan virtausta, joka kulkee läpi eri virtausyksiköiden, eli prosessivaiheiden. Virtaustehokkuus pyritään pitämään jatkuvasti käynnissä, eli tuotetta pyritään jalostamaan koko ajan tuottamalla sille lisää arvoa asiakkaan näkökulmasta. [31]

Prosessilait auttavat ymmärtämään sitä, mikä estää organisaatiota pääsemästä maksimaaliseen virtaustehokkuuteen, eli tuotteen läpimenoaikaan. Niiden mukaan läpimenoaika kasvaa riippuen siitä, montako keskeneräistä virtausyksikköä prosessissa on, ja kuinka pitkä jaksoaika on. Prosesseissa syntyy usein pullonkauloja, jolloin tehtävät kasautuvat virtausyksiköissä, ja viivästyttävät myös tätä seuraavia virtausyksiköitä. Läpimenoaika kasvaa, mitä lähempänä ollaan maksimaalista käyttöastetta. [31]

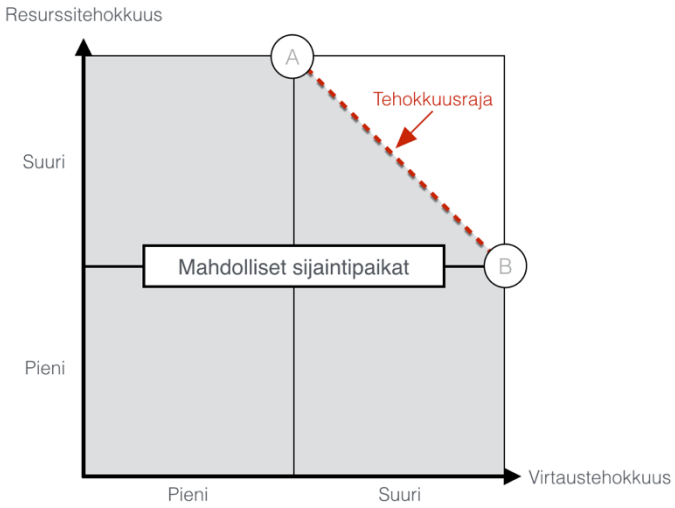
Tehokkuusmatriisia voidaan käyttää luokittelemaan organisaatio sen resurssi- ja virtaustehokkuuden perusteella. Alla olevassa kuvassa 6 on neljä aluetta, joille organisaatio voi sijoittua, ja tähti merkitsee optimaalista päämäärää.



Kuva 6 *Tehokkuusmatriisi [31, s. 100, 103]*

Tehokkuussaarekkeeseen sijoittuva organisaatio on keskittynyt resurssitehokkuuteen virtaustehokkuuden kustannuksella. Organisaatiossa kokonaiskuva voi olla kadonnut, koska asiakkaan tarpeiden täyttäminen on osaoptioitu eri yksiköille. Tehokkuus meressä tilanne on päin vastainen, eli organisaatio panostaa asiakkaan tarpeiden tyydyttämiseen, mikä edellyttää isoa määrää vapaata resurssien kapasiteettia. Joutomaalla organisaatio ei kykene luomaan tehokasta virtausta eikä optimoimaan resursseja, ja asiakas on tyytymätön. Ihannemaa on tavoitealue, kahden tehokkuusluokittelun tasapainoinen kompromissi. Tänne on vaikea päästä erityisesti vaihtelun ja siitä seuranneiden vaikutusten takia, joita avataan tarkemmin myöhemmin.

Tähden sijainti tehokkuusmatriisissa [kuva 7] tulisi olla tavoite, mutta käytännössä se on mahdotonta. Sinne sijoittuminen vaatii täydellistä tietoa asiakkaasta ja sen tarpeista, sekä täydellistä resurssien joustavuutta. Virheitä ei saa koskaan tulla, eikä työntekijät sairastua, eikä tekniikka saa pettää. Vaihtelu sanelee tehokkuusrajan, joka rajaa organisaation maksimaalisen tavoitetason. Tehokkuusrajan yläpuolelle pääsy on mahdotonta. Alla olevassa kuvassa 7 on havainnollistettu tehokkuusrajaa tehokkuusmatriisissa.



***Kuva 7** Tehokkuusmatriisin tehokkuusraja [31, s. 105]*

Organisaation sijoittuminen matriisiin on riippuvainen resurssitehokkuuden (A) ja virtaustehokkuuden (B) arvottamisesta. Myös organisaation strategia ja toimiala vaikuttavat sijoittumiseen tehokkuusrajalle. Organisaation menestystä ja tehokkuutta mittaa sen kyky mukautua vaihteluun. [31]

3.2.2.2 Vaihtelu

Vaihtelulla on suuri merkitys resurssi- ja virtaustehokkuuteen. Vaihtelua esiintyy ajassa, joka eri virtausyksiköiltä kuluu prosessin läpikäymiseen tai prosessin saapumiseen. Tällöin mahdollistuu pullonkaulojen muodostuminen. Vaihtelua ei voi täysin poistaa prosesseista. Etenkin, kun prosessissa virtausyksiköt ovat ihmisiä, aiheutuu vaihtelua luonnostaan esimerkiksi sairauspoissaolojen muodossa. Laatuongelmat aiheuttavat myös vaihtelua, sillä virhe tulee aina korjata, ja tämä viivästyttää myöhempää tuotantoa. Joskus prosessin eteneminen keskeytyy, koska asiakkaan palaute ei ole saapunut ajoissa. [31]

3.2.2.3 Tehottomuuden lähteet

On inhimillistä, että toissijaisia tarpeita syntyy prosessien aikana, kun vaatimukset liian monen asian yhtäaikaisesta tekemisestä lisääntyvät. Kun palveluyrityksessä henkilöstä joutuu käsittelemään suurta määrää asiakkaita samaan aikaan, saattaa yksittäinen asiakas kokea jäävänsä varjoon. Henkilökunta jää asiakkaalle persoonattomaksi, ja organisaation tuottama asiakaskokemus (*engl. cutomer experience, CX*) heikkenee. Mitä enemmän samassa prosessissa on asiakkaita, sitä vaikeampaa on saada asiakas tuntemaan, että hänestä välitetään ja hän on ainutlaatuinen. Pettynyt ja turhautunut asiakas on vaikea saada hyvälle tuulelle ja voittaa takaisin puolelle. Tämä vaatii lisäresursseja, ja aiheuttaa toissijaisia tehtäviä. Ihmisen aivot kykenet muistamaan noin 5-9 asiaa samanaikaisesti. Tämän jälkeen ihminen alkaa unohdella ja tehdä virheitä. [31]

Tarve monen asian yhtäaikaiselle hoitamiselle on peräisin resurssitehokkuuteen keskittymisestä, jolloin organisaatiolle on normaalia, että kaikilla on työtä tehtävänään jatkuvasti. Tämä aiheuttaa keskeneräisiä ja viivästyneitä virtausyksiköitä, eli vaihtelua, ja läpimenoajan kasvua. Ihmisissä usean virtausyksikön samanaikainen hoitaminen aiheuttaa turhautumista ja stressiä, kun tekijästä tuntuu, että asioiden hallinta karkaa käsistä. Stressi taas aiheuttaa kokonaiskuvan heikkenemistä ja ongelmien sivuuttamista. Monen tehtävän samanaikainen hoitaminen on henkinen haaste, koska huomio joudutaan siirtämään jatkuvasti asiasta toiseen. Mitä vähemmän ihmisellä on suoritettavia tehtäviä samanaikaisesti, sitä helpompaa keskittyminen on Yritykset ovat perinteisesti korjanneet tilanteen investoimalla lisäresursseihin, kehittämällä rutiineja ja muokkaamalla rakenteita, jotta kasautuneet tehtävät saadaan hallittua. Syntyy siis toissijaisia tarpeita. [31]

Kun asiakkaan ensisijaista tarvetta ei onnistuta tyydyttämään ajoissa pitkien läpimenoaikojen takia, voi syntyä toissijaisia tarpeita, joita ei alun perin ollut olemassa. Pitkät läpimenoajat turhauttavat, huolestuttavat ja pitkästyttävät, kun asioiden etenemistä joudutaan odottamaan. Odottaminen taas laskee ihmisten tarmokkuutta, inspiraatiota ja motivaatiota. Viivästyksiä syntyy etenkin, kun tehtävä joudutaan aloittamaan alusta. Uudelleen aloitettavia asioita ovat esimerkiksi asioiden uudelleen selittäminen kokouksesta myöhästyneelle henkilölle. Näiden asioiden korjaamisesta syntyy jälleen toissijaisia tarpeita joko projektin tai koko organisaation tasolla. [31]

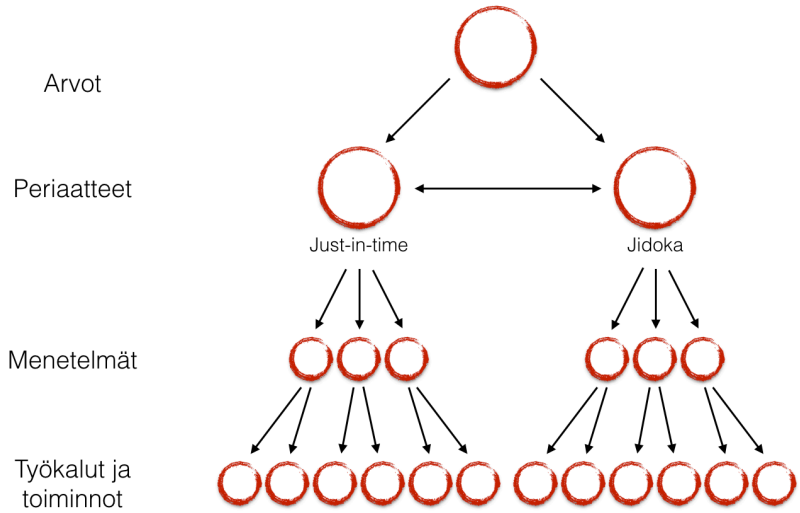
3.2.2.4 Toissijaiset tarpeet ja tehokkuusparadoksi

Lisätyöt itsessään aiheuttavat myös toissijaisia tarpeita, eli syntyy ketjureaktio. Toissijaiset tarpeet luetaan arvoa tuottamattomiin toimintoihin, eli ne ovat hukkaa, josta Leanissa pyritään pääsemään eroon. Toissijaisia tarpeita ei olisi, jos ensisijainen tarve olisi saatu tyydytettyä. Organisaatiot eivät aina ymmärrä, että lisätyö on hävikin muoto. [31]

Tehokkuusparadoksi tarkoittaa, että resursseja hukataan sekä yksilö-, että organisaatiotasolla. Kummankin kohdalla voidaan kysyä, paljonko lisätyötä tuli tehtyä työpäivän aikana. Tehokkuusparadoksin selittää lisätyö, ja ratkaisu löytyy virtaustehokkuuden paremmasta huomioimisesta, jolloin turha työ saadaan karsittua ja resursseja vapautettua, ja pystytään keskittymään arvon tuottamiseen paremmin. Käytännössä töiden uudelleen aloittaminen vältetään sillä, että samanaikaisesti hoidettavia tehtäviä vähennetään, ja töiden siirtäminen eteenpäin tapahtuu mahdollisimman joustavasti. Virtauksen pitää täten olla jokaiselle jatkuvasti näkyvissä, ja vastuu kokonaisuudesta kannetaan yhdessä. [31]

3.2.3 Leanin määritteleminen yrityksessä

Ihmiset pitävät Leania ratkaisuna kaikkiin käytännön ongelmiin organisaatiossa ja prosesseissa, ja sitä pyritään määrittelemään tavoilla, joita ei pystytä osoittamaan vääriksi. Tämän takia on tärkeää ymmärtää, mistä Leanissa on oikeasti kyse. Lean voidaan määritellä monella abstraktiotasolla, jotka poikkeavat selkeästi toisistaan. [31] Tätä on havainnollistettu alla olevassa kuvassa 8.



Kuva 8 *Lean eri abstraktiotasoilla [31]*

Organisaation arvot määrittelevät sen, miten toimitaan ja mitä organisaatio haluaa olla. Arvojen avulla ilmaistaan, millaisia meidän tulisi olla suhteessa tavoitteeseen. Esimerkiksi asiakkaan tarpeiden täyttäminen on arvo, joka luo edellytykset, että prosessista saadaan ulos juuri sitä, mitä asiakas halusi. Arvot taas vaikuttavat periaatteisiin, ja periaatteet päätöksentekoon ja asioiden priorisointiin. Periaatteet ovat tulosta arvojen seuraamisesta. Jos periaatteina olisivat Toyotan mallin mukaan Just-in-Time ja Jidoka, niiden tarkoitus on ehkäistä, tunnistaa ja eliminoida asiat, jotka häiritsevät virtausta. [31]

Menetelmät pyritään määrittelemään ja vakioimaan prosesseille koko yrityksessä. Vakioiduista menetelmistä heijastuu periaatteiden joustavuus tilannekohtaisesti. Vakiointi on yksi tärkeimmistä menetelmistä muun muassa virheiden havaitsemiseen. Menetelmiä toteutetaan työkalujen ja toimintojen avulla. Ne ovat konkreettisin ja käytännöllisin abstraktiotaso, joka ihmisten on kaikkein helpoin ymmärtää ja tunnistaa. Työkalu voi olla esimerkiksi visualisointitaulu, kuten Kanban-taulu, jolta organisaatio valvoo virtauksen etenemistä. Jos Lean määriteltäisiin vain tällä tasolla, olisi määritelmä liian kapea ja rajoittunut, koska näin käytännönläheiset keinot soveltuvat vain tiettyihin toimialoihin. Huomiota ei tulisi kiinnittää työkaluihin ja toimintoihin, eikä niistä saa tulla

itse tarkoitus. Huomio tulee kiinnittää tavoitteisiin, jolloin työkaluja ja menetelmiä voidaan käyttää joustavasti. [31]

Tavoitteiden tulisi olla osa liiketoiminta- ja toimintastrategiaa, jotta organisaatio kykenee rajaamaan ne asiakkaiden tarpeet, joita se pyrkii täyttämään. Käytännössä organisaatio määrittelee sen laadun, kokemuksen ja kustannukset, joita se asiakkaalle tuottaa. Toimintastrategiassa tulisi siis määritellä Lean korkealla tasolla, jotta sitä voidaan soveltaa eri konteksteissa. Kaiken toiminnan tulee kytkeytyä organisaation tavoitteisiin, ei menetelmiin tai työkaluihin. Virtaustehokkuus ei saa olla organisaation ainoa tavoite, vaan tärkeää on ymmärtää iso kuva, ja syy Lean menetelmän mukaan ottamiselle, mikä vaatii organisaatiolta omien prosessien kriittistä tarkastelua. [31]

Kun hierarkian eri tasot hahmotetaan keinoina, on helpompi ymmärtää, miten ne liittyvät toisiinsa. Hierarkian tasojen hahmottamisen lisäksi organisaation on ymmärrettävä, että Lean on dynaaminen tila. Aina on parantamisen varaa. Käytännössä organisaatio voi kuitenkin ajatella, että kun toimintastrategia on saavutettu, on tavoitekin tällöin saavutettu. [31]

3.3 Kanban-lähestymistapa

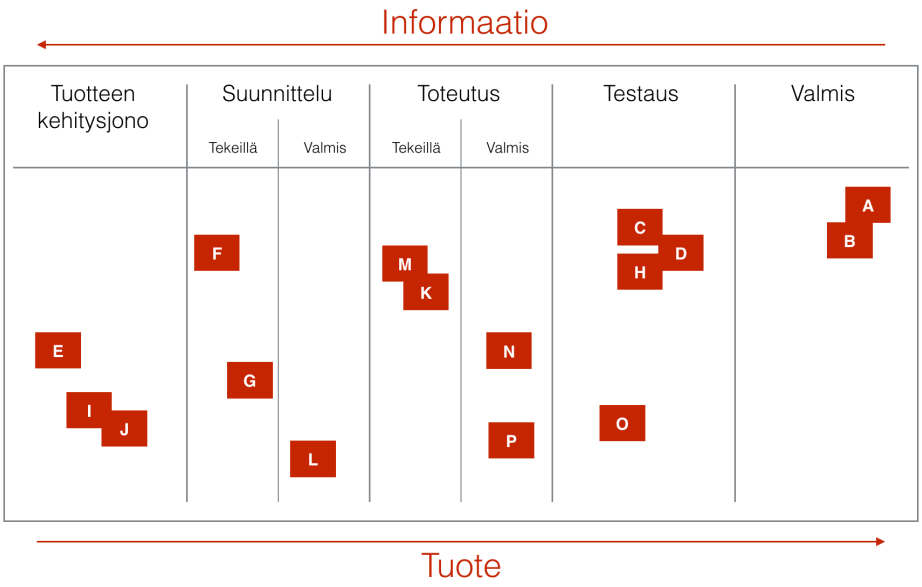
Kanban on yksi Lean-ajattelun toimintapa, joka kehitettiin Toyotalla 1953 Taiichi Ohnon toimesta visualisoimaan kaikille yksiköille virtauksen etenemistä. Kanbania voidaan käyttää työkaluna myös Scrumissa, tai sitä voidaan käyttää itse menetelmänä. [33, 80]

Kanban-sääntöjä on käännetty monella eri tavalla riippuen menetelmän käyttöympäristöstä ja -tarkoituksesta. Alkuperäisessä kokoelmassa säännöt nojautuvatkin tuotantoprosessin ja tuotantolinjan hallinnointiin:

- Älä toimita huonolaatuista tavaraa eteenpäin seuraavalle prosessille,
- Seuraava prosessi tilaa vain niin paljon kuin se tarvitsee,
- Toimita vain se määrä mikä on tilattu,
- Tasoita tuotanto,
- Kanban on työkalu hienosäätöön,
- Stabilisoi ja rationalisoi prosessi. [29]

Näitä sääntöjä voidaan helposti soveltaa ohjelmistokehityksen tarpeisiin. Työnkulku visualisoidaan pilkkomalla työt sopivan kokoisiin tehtäviin ja kirjaamalla ne erillisille korteille, ja kiinnittämällä ne Kanban-tauluun, joka on jaettu virtausprosessin mukaisiin vaiheisiin sarakkeittain. Jokaiselle sarakkeelle määritetään WIP (*engl. Work in Process*), eli suurin määrää tehtäviä, joita kyseisessä sarakkeessa saa kerrallaan olla, jotta töitä ei kasaannu liikaa yhdelle yksikölle ja näin muodostu pullonkauloja virtaukseen. Lisäksi kirjataan tehtävien läpimenoaika (*engl. lead time*), eli keskimääräinen aika, joka menee

yhden tehtävän valmistumiseen. Läpimenoaikojen kirjaamista voidaan jatkossa käyttää prosessin optimoimiseen ja ennustettavuuden parantamiseen. [79]



Kuva 9 *Esimerkki Kanban-taulusta*

Kanbanin työkaluista tärkein on siis taulu (*engl. board*), jota on havainnollistettu yllä olevalla kuvalla 9. Kehitystiimi määrittelee taulun lopulliset sarakkeet sen mukaan, mitä vaiheita kyseiseen prosessiin on otettu mukaan. Tärkeää on laittaa tauluun jokainen steppi tai vaihe, joka on mukana prosessissa, jotta koko kehitystiimille on näkyvää missä kohtaa prosessia asiat etenevät. Korteille kirjoitetaan tehtävä, joka pitää sisällään vaaditun toiminnallisuuden kuvauksen. Tehtäviä kuljetetaan taululla vaiheesta toiseen sitä mukaa, kun ne edistyvät. Kanban-taulu mahdollistaa työn edistymisen, koska se ei vaadi Scrumin kaltaista sprint-sykliä eikä virallista sitoutumista. [7] Kanbanin ja Scrumin ero on siinä, ettei Kanbanissa oli kehityssprinttejä, vaan suunnittelupalavereja pidetään vain tarvittaessa, ja ohjelmistosta julkaistaan uusi versio heti, kun pienin arvoa tuottava ominaisuusjoukko (*engl. Minimum viable product, MVP*) valmistuu. Näin ollen Kanban-taulu tyhjenee vasta, kun lopullinen, valmis tuote on saatu julkaistua. Tässä korostuu se, että valmis-tila on etukäteen määriteltävä. [79]

Kanban on hyvin kevyt työkalu, joka sopii parhaiten ympäristöön, jossa kehitystiimi on valmiiksi avoin, yhteistyökykyinen ja kommunikoi hyvin, eikä kulttuuri tarvitse koulutusta näiden tai ketterien menetelmien suhteen. [7]

4. SUUNNITTELU INTEGROITUNA KETTERYYTEEN

Kaikki ketterät menetelmät perustuvat alun perin Ketterän ohjelmistokehityksen julistuksen määrittelemiin arvoihin ja periaatteisiin, ja niiden toteuttamiseen käytetään erilaisia tapoja ja työkaluja. Käytännössä projektit edelleen kamppailevat lukkoon lyötyjen virstanpylväiden (*engl. milestones*) kanssa, vaikka uusia työtapoja kehitetään jatkuvasti. Julistuksen tarkoitus olikin määritellä arvojärjestelmä, joka sallii kulttuuriympäristön luomisen, pystyy vastaamaan virstanpylvästilanteeseen, jossa tunnustetaan yksilöiden arvo tiimin jäseninä ja tuotetaan hyvä järjestelmä. [7]

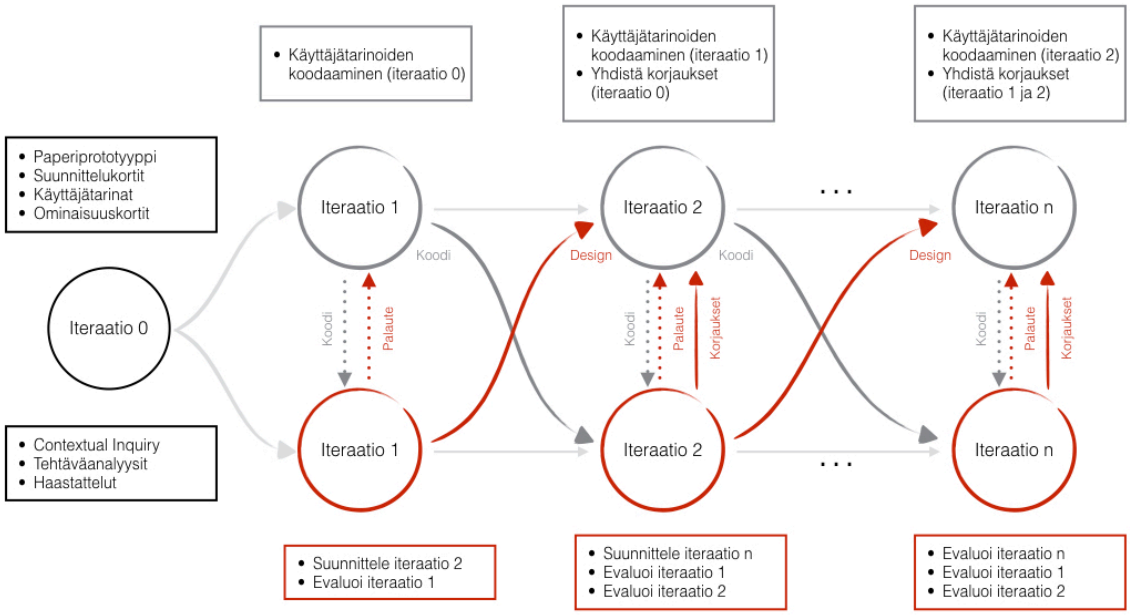
Ketterän ohjelmistokehityksen julistus ei käsittele käytettävyysnäkökulmaa osana ohjelmistokehitystä. [5]. Ei ole olemassa koottuja, standardoituja käytäntöjä, miten käyttäjäkeskeinen suunnittelu otetaan mukaan ketterään kontekstiin, siten että ketterät arvot ja periaatteet toteutuisivat. Lisäksi käyttäjäkeskeisyyden ja käyttäjäkokemuksen periaatteet ovat itsessään epäjohdonmukaisia, koska ne ovat useiden eri henkilöiden kirjoittamia. Tästä syystä useat yritykset kirjoittavatkin omat periaatteensa käyttäjäkeskeisen suunnittelun ja ketterien ohjelmistokehitysmenetelmien integroimiselle. Käyttäjäkeskeisen ja käyttäjäkokemuksen suunnittelun keskeisiin arvoihin kuuluu joka tapauksessa käyttäjän ja asiakkaan tarpeiden huomioonottaminen ja heidän osallistaminen käyttäjäkeskeiseen suunnitteluun. [7]

Käyttäjäkeskeisen suunnittelun ja ketterien ohjelmistokehitysmenetelmien välillä on filosofisia ja periaatteellisia eroja muun muassa evaluointimenetelmissä, kulttuurissa ja dokumentaatiossa. Ketterissä menetelmissä kehitys on hajotettu toiminnalliselle tasolle. Käyttöliittymää ja käyttäjäkokemusta tulisi kuitenkin kehittää kokonaisvaltaisesti. Vaiheistus voi vaarantaa suunnittelijan vision tuotteen käytettävyydestä, käyttöliittymästä, sen kokonaiskuvasta ja lopputuloksesta. [36] Käyttäjäkeskeisellä suunnittelulla ja ketterillä prosesseilla on myös yhteneväisyyksiä, kuten käyttäjien ja asiakkaiden tarpeiden huomioonottaminen ja heidän osallistaminen käyttäjäkeskeiseen suunnitteluun, sekä palautteen kerääminen. Sekä ketterien menetelmien että käyttäjäkeskeisen suunnittelun keskeisenä menetelmänä on ratkaisujen iterointi, vaikkakin ne iteroivat eri asioita. Käyttäjäkeskeisen suunnittelun iterointia tehdään korkeammalla abstraktiotasolla kuin ketterissä menetelmissä, joissa iterointi tapahtuu matalalla, toiminnallisella tasolla. Tästä syystä näiden kahden iteratiivisen menetelmän integraatio on koettu haasteelliseksi. Ohjelmistokehityksen käyttämä ketterä iteraatiomalli pakottaa käyttäjäkeskeisen suun-

nittelun mukautumaan näihin ensisijaisesti ohjelmistokehitystä varten luotuihin prosesseihin. [7, 13, 14]

4.1 Etupainotteinen suunnittelu

Vuonna 2007 Sy [39] esitteli prosessimallin, jolla käyttäjäkeskeinen suunnittelu voidaan tuoda osaksi ketterää viitekehystä työstämällä rinnakkain kahta polkua (*engl. dual track*). Alla ovessa kuvassa 10 on suomennettu version Syn kehittämästä mallista.



Kuva 10 *Syn suunnittelumalli [39]*

Syn mallissa vaatimusten kerääminen ja suunnittelu tehdään etupainotteisesti. Vaatimukset kerätään 1-2 iteraatiota ennen kehitystä, ja yksityiskohtainen suunnittelu vähintään yksi iteraatio ennen kehitystä. Ensimmäistä suunnitteluiteraatiota kutsutaan iteraatio 0:ksi, sprint 0:ksi tai nollaiteraatioksi. Ennen ensimmäistä toteutusiteraatiota suunnittelijoilla tulee olla ymmärrettynä tuotteen kokonaiskuva ja määriteltynä vaatimukset ja tavoitteet käyttäjätutkimusten perusteella. Kun kehitys aloittaa tuotteen toteutuksen, suunnittelijat työstävät seuraavaa asiaa ja evaluoivat edellistä iteraatiota. Näin suunnittelijat työskentelevät projektin aikana yhden iteraation toteutusta edellä. [39]

Brown pohti kirjassaan [7], että käytettävyystiimistä, projektin laajuudesta ja iteraatioiden pituudesta riippuen voidaan miettiä, onko tiimin parempi työskennellä yhtenäiseen tahtiin toteutustiimin kanssa, vai tehdä suunnittelu etupainotteisesti. Etupainotteinen suunnittelu antaa toki mahdollisuuden iteroida tehtyjä ratkaisuja, kerätä palautetta sekä reagoida suunnitteluvirheisiin ja -ongelmiin hyvissä ajoin. Jos käyttäjäkeskeinen suunnittelu taas tehdään toteutuksen kanssa samanaikaisesti, tulee suunnittelijan ja kehittäjän

työskennellä erittäin tiiviinä työparina rinnakkain. Tällöin suunnittelua voidaan tehdä todella pieni kokonaisuus kerrallaan, jolloin kokonaiskuvan hahmottaminen entisestään vaarantuu. Joka tapauksessa oleellista on, että suunnittelija ja kehittäjä kommunikoivat avoimesti koko prosessin ajan.

Fox et al. [14] huomauttavat, että todellisuudessa nollaiteraatiossa ei ole tarpeeksi aikaa, eikä mahdollisuutta toteuttaa kuin pieni määrä toiminnallisuuksia [14]. Kollman et al. [23] toteavat saman omassa tutkimuksessaan. Suunnittelun tekeminen kaksi viikkoa toteutuksen edellä saattaa olla liian lyhyt aika etenkin, kun kyseessä on monimutkaiset ja monitahoiset toiminnallisuudet. Heidän tutkimuksessaan tuli esiin myös, että nollaiteraatiota tulisi toistaan monta kertaa, koska priorisoinnit tulisi olla kunnossa ennen toteutusvaiheiden aloittamista. Nollaiteraatioiden lisääminen toteutuksen eri vaiheisiin ei välttämättä ratkaise ongelmaa, että tiimi kadottaa kokonaiskuvan tuotteesta. Kaksi viikkoa voi olla riittävä aika vaatimusten perusteiden ymmärtämiseen, mutta suunnitteluun liittyy lisäksi muun muassa käyttöliittymän ja käyttäjäkokemuksen suunnittelu. [23]

Kuusinen [25] huomioi, että ongelmat etupainotteisessa suunnittelussa liittyvät myös suunnittelijoiden ajan puutteeseen ja tehtävien määrään. Periaatteessa suunnittelijat työskentelevät Sy:n mallissa neljän päällekkäisen iteraation kanssa samanaikaisesti, sillä käyttäjätutkimukset voidaan tehdä kaksi iteraatiota edellä, varsinainen suunnittelu yhtä iteraatiota edellä, ja evaluointi suunnittelulle tehdään kehitysiteraation jälkeen. Tämä tarkoittaa neljän erillisen polun (*engl. track*) tehtävien tekoa samanaikaisesti. Brhel et al. ovat tutkimuksessaan [6] luokitelleet yleisimpiä käyttäjäkeskeiseen suunnitteluun liittyviä tehtäviä ketterissä ohjelmistoprojekteissa, jotka on koottu taulukkoon 2. Tehtävät on taulukossa luokiteltu käyttäjätutkimukseen, konseptointiin, suunnitteluun ja evaluointiin. Brhelin et al. mukaan ylimmällä rivillä on yleisimmin esiin tulleet suunnittelumenetelmät. Suunnittelijoiden tehtäviä ja rooleja on käsitelty tarkemmin kappaleessa 6.2 *Suunnittelijoiden tehtävät* ja kappaleessa 7.1. *Suunnittelijoiden rooli ja tehtävät ketterässä tiimissä*.

Taulukko 2 *Ketterien käyttäjäkeskeisten suunnittelumenetelmien luokittelu niiden esiintyvyyden osalta Brhelin tutkimuksessa [6].*

	Käyttäjätutkimus	Konseptointi	Suunnittelu	Evaluointi
Yleisin	Contextual inquiry	Käyttäjätarinat	Prototypointi	Käyttäjätestaus
	Tehtäväanalyysi	Toimintaohjeet (*engl. guidelines*)	Mock-upit	Asiantuntija-arvio
	Focus group	Skenaariot	Rautalangat	Asiakkaan tai käyttäjän osallis-taminen
	Haastattelut	Vaatimukset		Valmis-kriteerit
	Kyselyt	Priorisointi		
Harvinaisin		Persoonat		

4.2 BoB (Best of Both Worlds) -viitekehys

Vuonna 2015 Kuusinen julkaisi väitöskirjan [25], jossa hän esitteli BoB (*Best of Both Worlds*) -viitekehyksen parhaista käytännöistä integroida käyttäjäkeskeisen suunnittelu ketterään yritysohjelmistokehitykseen. Viitekehys tähtää laadun maksimointiin sekä ajan ja kustannusten minimoimiseen. BoB-viitekehys koostuu neljästä pääosasta, joita ovat

1. Agile UX –työn syötteet (*engl. input*) ja tulokset (*engl. output*),
2. mittarit tulosten, prosessin ja syötteiden evaluoimiseen
3. prosessi, jossa tulokset on luotu ja
4. aktiviteetit ja tehtävät, jotka toteuttavat kyseisen prosessin.

Tavoitteena viitekehyksellä on mahdollistaa ristiriidaton ohjelmistotuotanto, jolla on kunnollinen laatu ja laajuus (*engl. scope*). Lisäksi tavoitteena on olla kustannustehokas, esimerkiksi välttämällä ali- ja ylituotantoa, ja nopea. Viitekehyksellä halutaan mahdol-listaa tiimin työskentely kohti yhteistä ja yhdessä sovittua lopputuotosta maksimoimalla laatu, voitto ja tyytyväisyys samalla minimoiden kustannukset, aika ja riskit.

Syötteet ovat ohjelmistokehityksessä tämän viitekehyksen mukaan muun muassa tiimi, tehtävät, prosessi, työkalut ja käyttäjät. Näiden kaikkien tekijöiden syötteet muodostavat tuotokseksi yritysohjelmiston. Käyttäjät eroavat muista syötteistä siinä, että he myös

loppujen lopuksi päättävät, onko tuotos onnistunut. Muita kehityksen onnistumisen ja suorituskyvyn mittaamiseen käytettäviä tekijöitä ovat laatu, laajuus, kustannukset ja käytetty aika.

Projektin raaka visio tulisi olla valmiina aikaisessa vaiheessa, ja se pitää kommunikoida koko tiimille ymmärrettävästi, ja siitä pitää muistuttaa koko projektin ajan. Tavoitteena on saada koko projekti työskentelemään yhteistä visiota kohden. Suunnittelijan rooli on tässä vaiheessa erityisen tärkeä. Aikainen visio vähentää epävarmuutta, ja lisää selkeyttä projektin isoon kuvaan koko sen elinkaaren ajan. Visio saadaan muodostettua käyttäjätarpeiden ja käyttäjien niille antaman arvon perusteella, joiden selvittäminen edelleen kuuluu yleisesti ja ensisijaisesti suunnittelijoiden tehtäviin. Käyttäjätarpeiden väärinymmärtäminen voi johtaa väärän tuotteen kehittämiseen. Visio konkretisoidaan konseptiksi, ja käyttäjien tarpeilleen antamien arvojen realisointi tapahtuu suunnittelun ja kehitystyön avulla. Evaluointia voidaan tehdä koko projektin läpiviennin ajan, niin visiolle, konseptille, designille kuin toimivalle ohjelmistolle.

Kuusinen arvio käyttäjäkeskeisen suunnitteluprosessin itsessään liian tiukaksi ketteriin menetelmiin. Esimerkiksi perinpohjaista käyttäjätutkimusta harvoin ketterissä projekteissa tarvitaan, jos tiimiltä aidosti hyväksytään kokeilut ja mahdolliset virheet. Käyttäjätutkimusvaiheessa visio on vasta kehittymässä ja käyttäjien kanssa keskustellaan abstraktilla tasolla. Nollaiteraation sijaan isompien projektin alussa voi olla useampia suunnittelukeskeisiä iteraatioita, jolloin kehitys voi aloittaa ohjelmiston perusteiden rakentamisen. Etupainotteinen suunnittelutyö pitäisi kuitenkin minimoida, sillä aikainen visiointityö keskittyy ymmärtämään vain alleviivattuihin ilmiöihin. Kuusinen kehottaa aloittamaan suunnittelijoiden ja kehittäjien yhteistyön muutamilla lyhyillä työpajoilla, joissa käsitellään muun muassa käyttäjätarinoita ja perustetaan tuotteen kehitysjono, ja joiden välillä tiimi työskentelee pienin askelin kosketeltavaa ja näkyvää tuotetta kohden. Alkuvaiheessa tuote voi olla prototyyppi tai läpiklikkailtava malli. Kun tiimi ei kykene ratkaisemaan suunnitteluongelmaa, kehittäjät siirtyvät kehitysjonon seuraavaan tehtävään, ja suunnittelijat pyrkivät ratkaisemaan ongelman. Näin tehdään, kunnes ongelma on ratkaistu tai tehtävää on päätetty siirtää tuonnemmaksi. Tiimi pyrkii julkaisemaan ensimmäisen toimivan version tuotteesta mahdollisimman nopeasti. Evaluoinnin, iteroinnin ja palautteen keräämisen tärkeys korostuu ketterissä menetelmissä. Tiimi testaa iteraation tuotoksia, korjaa virheet tai siirtää ne seuraavaan iteraatioon. Iteraation jälkeen kehitysjonon prioriteetit tarkastetaan ja jonoa muokataan. On huomioitavaa, että elinkelpoisen idean validointi usein realisoituu vasta ohjelmiston tultua käyttäjien saataville.

5. TUTKIMUSPROSESSI

Tässä luvussa käydään läpi diplomityön tutkimustavoitteet ja käytetyt tutkimusmenetelmät. Tämän lisäksi esitellään tutkimusasetelma sekä aineiston käsittely ja analyysimenetelmät.

5.1 Tutkimuksen tavoitteet

Tutkimuksen tavoitteena on hyvien käytäntöjen, menetelmien ja toimintatapojen löytäminen, kun käyttäjäkeskeinen suunnittelu ja ketterä kehitystyö yhdistetään ohjelmistoprojektissa. Näiden pohjalta on ollut tavoitteena hahmotella viitekehys tai toimintaohje (*engl. guideline*) ketterän ohjelmistokehitysprosessin läpiviemiseksi käyttäjäkeskeisen suunnittelun näkökulmasta.

Diplomityölle asetettiin kolme tutkimuskysymystä:

1. Miten käyttäjäkeskeinen suunnittelu otetaan huomioon osana ketterää ohjelmistokehitysprosessia?
2. Mitkä käytännöt tukevat suunnittelu- ja kehitystyön integraatiota ketterässä ohjelmistokehitysprosessissa?
3. Mitkä suunnittelutehtävät koetaan tuottavan eniten hyötyä prosessin ja lopputuloksen kannalta?

Ensimmäinen tutkimuskysymys on alun perin asetettu muotoon: ”Miten yritys ottaa huomioon käyttäjäkeskeisen suunnittelun osana ketterää ohjelmistokehitysprosessia”. Tutkimuskysymys on asetettu tähän muotoon haastatteluja silmällä pitäen, mutta tutkimuksen kokonaistavoitteen takia ensimmäinen tutkimuskysymys muutettu myöhemmin nykyiseen muotoonsa.

Tutkimuksen tuloksia on esitelty erikseen haastattelusta luvussa 6 ja verkkoaineistoanalyysin osalta luvussa 7. Haastattelutulokset ja verkkoaineistoanalyysin tulokset on esitetty eri luvuissa, koska haastatteluiden tavoitteena oli löytää nykyisiä todellisia käytäntöjä ja toimintatapoja ohjelmistoyrityksissä, ja verkkoaineistoanalyysin tarkoituksena oli laajentaa haastattelutuloksista saatua näkökulmaa. Verkkoaineisto antaa haastattelutuloksille myös vertailukohdan, ja auttaa haastattelutulosten hyvyyden ja validiteetin arvioinnissa. Molempien menetelmien tulokset on vedetty yhteen luvussa 8, jossa kootaan myös vastaukset tutkimuskysymyksiin.

5.2 Tutkimusmenetelmät

Yrityksiin kohdistuvassa tutkimuksessa menetelmänä käytettiin haastatteluja. Haastattelut vaihtelivat yksilöhaastatteluista ryhmähaastatteluihin riippuen siitä, miten yrityksellä oli mahdollisuus tarjota henkilöitä haastateltavaksi työajan puitteissa. Valittujen yritysten taustatiedot selvitettiin verkosta löytyvien yritystietojen avulla, ja ne on esitelty kappaleessa 5.3

Haastattelut olivat teemahaastatteluja. Haastattelukysymyksiä oli laadittu melko korkealla tasolla teemoittain, ja haastattelutilanteissa oli mahdollista pomppia teemoista toiseen. Haastattelussa keskityttiin nimenomaan suunnittelijan näkökulmaan työskennellä ketterässä ohjelmistokehitysprojektissa. Haastattelukysymykset löytyvät liitteestä A.

Haastattelujen rinnalle menetelmäksi valittiin verkkoaineistoanalyysi, jotta tulosten näkökulmaa saatiin laajennettua, ja kerättyä tuloksia ohjelmistoyritysten todellisista käytännöistä. Verkkoaineistoanalyysin avulla haastattelutuloksia voitiin validoida ja analysoida perusteellisemmin, ja saatuja tuloksia voitiin käyttää vertailukohtana haastattelutuloksille.

5.3 Haastattelututkimuksen tutkimusasetelma

Ennen haastattelujen aloittamista tehtiin Tampereen teknillisen yliopiston kanssa vierailijatutkimussopimus, jolloin haastattelijaa sitoo kaikki samat salassapitovelvollisuudet kuin yliopiston tutkijoita. Haastateltaville annettiin mahdollisuus nähdä allekirjoitettu vierailijatutkimussopimus, ja heille kerrottiin suullisesti sopimuksen sisällöstä ennen haastattelujen alkua.

Tutkimuskohteena oleva yritykset ovat kohtalaisen nuoria ohjelmistokehitystyöhön keskittyneitä osakeyhtiöitä. Kaikilla tutkimukseen valituilla yrityksillä on käyttäjäkeskeistä suunnittelua mukana projektityössä, ja kaikki valitut yritykset käyttävät projektinhallinnassa ketteränä ohjelmistokehitysmenetelmänä Scrumia, Leania tai Kanbania vaihtelevilla painotuksilla. Alla olevassa taulukossa 3 on koottuna yritysten tiedot, jotka ovat osallistuneet haastattelututkimukseen.

Taulukko 3 *Haastateltujen yritysten perustiedot järjestettynä perustamisvuoden mukaan. [50]*

Tunniste	Perustamisvuosi	Toimipaikan henkilöstöluokka	Toimialakohtainen luokitus	Liikevaihtoluokka
H1	1996	250 – 499	Ohjelmistojen suunnittelu ja valmistus	20-100M €
H2	2000	100 – 249	Ohjelmistojen suunnittelu ja valmistus	20-100M €
H3	2000	100 – 249	Tietojenkäsittely, palvelintilan vuokraus ja niihin liittyvät palvelut	20-100M €
H4	2001	100 – 249	Ohjelmistojen suunnittelu ja valmistus	10-20M €
H5	2008	< 50	Ohjelmistojen suunnittelu ja valmistus	2-10M €
H6	2014	100 - 249	Ohjelmistojen suunnittelu ja valmistus	2-10M €

Haastateltavat olivat yrityksen suunnittelijoita eri suunnitteluosaamispainotuksilla. Yksilöhaastatteluja tutkimuksessa oli kaksi kappaletta, muut olivat ryhmähaastatteluja, joihin osallistui kahdesta neljään suunnittelijaa. Yhteensä haastatteluihin osallistui 13 suunnittelijaa. Haastattelut kestivät osallistujamäärästä riippuen tunnista puoleentoista tuntiin.

Haastattelut tallennettiin äänitallentimella, jonka lisäksi haastattelija teki haastattelujen aikana muistiinpanoja. Haastateltavat saivat haastattelun aikana piirustustehtävän, jonka haastateltavat haastattelun aikana selittivät auki. Piirustustehtävä tuki yritysten iteraatioprosessien ja käytäntöjen ymmärtämistä. Kuvat 11 ja 12 ovat otteita kyseisestä tehtävästä.

5.4 Aineiston käsittely ja analyysimenetelmä

Haastattelujen yhteydessä haastateltaville yrityksille annettiin tunnistetieto, jolla haastatteluäänitteet ja haastattelujen yhteydessä tehdyt piirustukset merkittiin. Luvussa 6 tunnistetietoihin ei ole viitattu, jotta yrityksiä ja tuloksia ei voitaisi yhdistää ja yritysten yksityisyys säilyisi. Haastatteluissa äänitetyt tallenteet litteroitiin haastattelujen jälkeen, ja sisältö kirjoitettiin tekstiksi. Haastateltavien hyviä kommentteja kerättiin taulukkoon aihealueittain. Tämän jälkeen datasta kerättiin tärkeimmät huomiot post-it -lapuille ja luokiteltiin ennaltamääriteltujen aihealueiden mukaan seinälle. Luokat liittyivät muun

muassa suunnittelijoiden ja kehittäjien väliseen työnjakoon, kommunikointiin ja iteraatioprosessin eri vaiheisiin, sekä käytettyihin menetelmiin. Lopulliset luokat vastaavat melko pitkälti luvun 6 kappalejakoa.

Haastattelujen analyysin jälkeen alettiin käydä läpi verkkoaineistoa. Koko diplomityön tekemisen ajan oli kerätty talteen aiheeseen liittyviä blogeja, artikkeleita ja keskusteluja. Verkkoaineistonanalyysin aluksi tehtiin vielä tarkemmin haku viimeisimpien aineistojen osalta. Lopulta verkkoaineistosta päätyi mukaan diplomityöhön 23 blogia, artikkelia tai keskustelua, sen perusteella, kuinka hyvin niissä vastattiin tutkimuskysymyksin. Lähteiden luotettavuus on pyritty takaamaan valitsemalla lähteet sivustoilta, jotka ovat keskittyneet käytettävyyteen, ketteriin menetelmiin tai näiden yhdistelmään, kuten esimerkiksi Agile Modeling [46], Nielsen Norman Group [61] ja UXmatters [65]. Mukana on myös joitain Medium [59] -artikkelialustalta löydettyjä alan ammattilaisten kirjoituksia, joiden ammattimaisuus on varmistettu tarkistamalla kirjoittajan ammatillinen tausta käyttäen ensisijaisesti LinkedIn-alustaa [56]. Valittu aineisto luettiin läpi, jonka aikana tutkija teki aineistosta muistiinpanoja, ja taulukoi niitä seuraten melko pitkälti haastattelujen yhteydessä tehtyä luokittelua.

Tulosten koontia ja lopullista raportointia varten tutkija yhdisti seinälle kerätyt haastattelutulokset ja taulukoidut verkkoaineistotulokset, ja vertaili näiden tuloksia keskenään.

6. HAASTATTELUTULOSTEN TARKASTELU

Tuloksissa on käsitelty sekä koko ketterää prosessia, mutta pääfokuksessa on suunnittelijoille yleiseensä kohdentuneet tehtävät ja vastuualueet, että iteratiivinen prosessi suunnittelijoiden näkökulmasta. Tärkeimmät tulokset on korostettu lihavoinnilla.

6.1 Suunnittelu- ja kehitystiimi

Kehitystiimin kokoonpano vaihtelee riippuen projektin suunnittelu- ja teknisyyspainotuksesta. Yritykset pyrkivät kasaamaan tiimit ensisijaisesti vapaana olevista, motivoituneista kompetensseista. **Suunnittelijoita tiimissä on yhdestä kahteen**, ja kehittäjiä yhdestä kuuteen. Mukana on myös projektipäällikkö tai muu asiakaskontaktihenkilö. Projekteissa harvemmin on tuoteomistajaa. Jos tuoteomistaja on, se on vaihtelevasti asiakkaan puolen edustaja tai kehitystiimin suunnittelija.

Roolien ja tehtävänjakojen osalta yrityksissä on vain vähän vaihtelua. Kahdessa yrityksessä suunnittelijat ovat selkeästi vain suunnittelijoita. Muissa yrityksissä **roolien ja tehtäväjaon osalta korostetaan joustavuutta ja tiimin itseohjautuvuutta.** Näissä **tehtävät ja roolit muodostuvat tiimin jäsenten ydinkompetenssialueiden ympärille**, eikä tekijöitä haluta lokeroida rooleihin, koska tekijöillä voi olla kompetenssiosaamista ja kiinnostusta yli roolirajojen. Yksi yritys pyrkii lokeroimisen ja vahvojen työparien muodostumisen välttämisellä aktiivisesti pitämään tiimin yhtenäisenä ja **osallistamaan kaikkia tiimin jäseniä kaikkiin tehtäviin**. Heillä **koko tiimi osallistuu asiakastapaamisiin** alusta lähtien, koska kokonaiskuva hahmottuu näin paremmin koko tiimille. Vaikka tiimit ovat roolien ja tehtävien suhteen itseohjautuvia, asiakkaat ovat tottuneet ostamaan ja näkemään rooleja, ja kommunikoimaan roolien kautta.

6.2 Suunnittelijoiden tehtävät

Projekteihin pyritään sijoittamaan kaksi suunnittelijaa tai suunnittelukompetenssia omaavaa henkilöä. Suunnittelijoiden kokonaismäärä ja kompetenssien painotus riippuvat projektin koosta ja tavoitteesta. **Kaksi suunnittelijaa takaa suunnittelulle laadun ja auttaa luomaan innovatiivisia ratkaisuja**, kun suunnittelijat sparraavat ja validoivat toistensa aikaansaan käyttäjälähtöisestä näkökulmasta. Tästä syystä yhden suunnittelijan projekti saatetaan allokoida kahdelle suunnittelijalle suhteella 80:20, jotta pääsuunnittelijalla on tukea saatavilla.

Suunnittelijoiden työ koostuu erilaisista tehtävistä riippuen, millaista kompetenssiosaamista tekijöillä on. Alla olevaan taulukkoon 4 on koottu haastatteluissa esiin tulleita suunnittelijoiden työtehtäviä.

Taulukko 4 *Suunnittelijalle kuuluvat tehtävät yrityksissä.*

	H1	H2	H3	H4	H5	H6
Myynnin tuki	✓		✓			
Liikejohdon konsultointi	✓					✓
Konseptointi						✓*
Palvelumuotoilu			✓*	✓*	✓	
Sisällöntuotanto						✓
Esiselvitys ja käyttäjätutkimus	✓	✓	✓	✓	✓	✓
Vaatimusmäärittely	✓		✓			✓
Käyttäjäkokemussuunnittelu (ml. interaktio-, visuaalinen ja graafinen suunnittelu)	✓	✓	✓	✓	✓	✓
Front end -kehitys	✓ *	✓ *	✓ *		✓ **	✓ *
Käytettävyystestaus	✓ **	✓ **	✓	✓ **	✓ **	✓
Dokumentointi			✓ **			
Tuoteomistajan (kaltainen) rooli / projektinhallintaa	✓		✓			
Kehitysjonon hallinta ja priorisointi	✓		✓		✓	

**Osaamisesta ja kiinnostuksesta riippuen*
***Harvoin tai todella harvoin*

Usein suunnittelijoiden työ alkaa myyntivaiheesta, jolloin he toimivat myyntitilaisuuksissa asiantuntijan roolissa, ja avustavat tarjousten teossa. Myyntivaiheeseen osallistuminen antaa suunnittelijalle mahdollisuuden tutustua asiakkuuteen ja toimialaan. Puolet haastateltavista yrityksistä osallistaa suunnittelijaa aktiivisesti myyntiin.

Suunnittelijan työ painottuu projektin alkuun esitutkimuksen, konseptoinnin ja vaatimusmäärittelyiden muodossa. **Hyvällä esiselvityksellä varmistetaan, että tehdään oikeaa asiaa, ymmärretään sidosryhmien tarpeet ja toiveet perusteluineen, ymmärretään mahdollisuudet ja rajoitteet, sekä voidaan vähentää testauksen tarvetta iteraatioprosessin aikana.** Esiselvityksen ja vaatimusmäärittelyiden pohjalta suunnittelijoiden vastuulla on luoda persoonat ja käyttötapaukset, jotka sijoitetaan ja priorisoidaan kehitysjonoon.

Alussa suunnittelijan tulee ymmärtää kokonaisuus. Järjestelmää suunniteltaessa aloitetaan yleensä navigaatiosta, jonka suunnittelu vaatii sisällön ja hierarkian ymmärtämistä. **Ison kuvan ymmärtäminen vie aikaa, joten suunnittelutyö aloitetaan monesti viikkoa tai kahta ennen kehityksen alkua.** Kokonaisuutta aletaan hahmottaa kehitysjonolle sijoitettujen käyttötapausten avulla. Projektin aikana kehitysjono sisältää sekä tulevia että suunnitteluun palautuvia käyttötapauksia. Molempien työstäminen samanaikaisesti vaatii suunnittelijalta organisointia ja suunnitelmallisuutta.

Suunnittelijat toimivat lopputuotteen laadunvarmistajina, kokonaiskuvaa ylläpitävänä voimana, ja henkilöinä, jotka varmistavat liiketoiminnan jatkumisen myös projektin jälkeen kokoamalla projektin aikana muun muassa jatkokehitysajatuksia. Laadun varmistus vaatisi käyttäjätestausta, mutta suurin osa yrityksistä on luopunut siitä. Suunnittelijat testaavat itse suunnittelemansa toiminnallisuudet, kun ne ovat valmiita.

6.3 Suunnittelijoiden kokonaistoimenkuva

Suunnittelijoilla saattaa olla useampia projekteja samaan aikaan tekeillä. Yleensä yksi projekteista on suurempi ja muut pienempiä, tai suunnittelijoilla on suuremman projektin ohella avustavia tehtäviä muissa projekteissa. Eräs suunnittelija tosin mainitsi, että hänellä saattaa olla yksi suurempi projekti ja neljä pienempää päällekkäin. Muissa yrityksissä suunnittelijoille allokoidaan keskimäärin kaksi tai kolme yhtäaikaista projektia. Ajoittain suunnittelijoiden työkuorma on valtava, ja heitä stressaa mahduttaa pahimmillaan yhden päivän työt kahteen tuntiin. **Monen projektin yhtäaikainen tekeminen muodostuu helposti pullonkaulaksi**, koska työt kasaantuvat. Kiireen aikana suunnittelija ei ole kehittäjien tavoitettavissa tarpeeksi nopeasti, jolloin kehittäjien työt odottavat. **Suunnittelijat eivät voi allokoida tiettyjä päiviä tietyille projekteille**, vaan heidän tulee olla kaikkien tavoitettavissa joka päivä. Tästä syystä suunnittelijoiden päivät helposti pirstoutuvat ja tiimin sisäinen kommunikaatio heikkenee merkittävästi. **Työmäärän vaihtelua on vaikea arvioida etukäteen** jatkuvien muutosten ja parannusten takia, joka johtaa liian moneen samanaikaiseen projektiin, kun resursoijat havaitsevat suunnittelijalla olevan kalenterissa tilaa.

Yksi yritys kohdentaa suunnittelijoille asiakkuuksia, ja asiakkuuden sisällä voi olla monta projektia. Tämä helpottaa suunnittelijoiden työtä, koska suunnittelijoiden ei tarvitse olla useassa lokaatiossa yhtä aikaa. Näissä yrityksissä on tapana työskennellä asiakkaan tiloissa, jolloin suunnittelijan on helpompi työstää useaa saman asiakkaan projektia samaan aikaa. Haasteena tällaisessa asetelmassa on suunnittelijan omat intressit ja motivaatio tehdä vain yhdellä toimialalla tai vain yhdellä asiakkaalla projekteja. Suunnittelijat vaihtavat pitkäaikaisia asiakkuuksia ajoittain pitääkseen yllä motivaatiota ja ammattitaitoa, sekä oppiakseen uutta.

Asiakkaat näkevät usein suunnittelijat pelkkinä määrittelijöinä, jotka leikataan projektista pois, kun kokonaisuus on hahmotettu ja rautalankamallit toimitettu kehitykselle. Yhdessä yrityksessä myös sisäisesti ajatellaan näin. Kyseisen yrityksen suunnittelijat kokevat olevansa tiimin ulkopuolinen osa, vaikka samaan aikaan he kokevat olevansa vastuussa lopputuloksesta. Muutoksia ei myöskään hyväksytetä suunnittelijoilla, jolloin suunnittelijoilla on vaikeuksia pysyä projektin perässä. Suunnittelijat kokevat tilanteen stressaavana. Muissa yrityksissä **suunnittelijat tai suunnittelukompetenssit ovat mukana koko projektin ajan**, vaikka työt vähenisivät loppua kohden. **Suunnittelijoiden määrää voidaan vähentää**, kun suurin työkuorma on tehty.

6.4 Kommunikaatio tiimissä ja asiakkaan kanssa

Haastatteluissa korostui kommunikaation tärkeys. **Pahimmat ongelmat muodostuvat kommunikaation puutteesta ja toisaalta persoonien tavoissa kommunikoida.** Kommunikaation tärkeys ei koskenut vain tiimiä, vaan myös asiakasta. Yritykset ovat käytännössä todenneet, että parasta mahdollista ymmärrystä asioista ei saavuteta sillä, että tuoteomistaja tai projektipäällikkö hallinnoi tiimin ja asiakkaan välistä kommunikaatiota, vaan **tiimin pitää voida itse kommunikoida asiakkaan kanssa suoraan, ja esittää tarkentaviakin kysymyksiä.** Kommunikaatio ei saa olla riippuvainen kummankaan osapuolen henkilöiden asemasta, työvuosista, roolista, eikä statuksesta.

Paras ja tehokkain kommunikointitapa on yksimielisesti **kasvokkain käytävä kommunikaatio**, joka on avointa, suoraa, luontevaa, välitöntä, reaaliaikaista ja takaa parhaan mahdollisen ymmärryksen kaikille osapuolille. Hyvä kommunikaation näkyy synergiana. Kun omissa töissä on ongelmia ja ajatukset jumissa, välitön kommunikaatio helpottaa avun pyytämistä. **Välittömyys ja avoimuus näkyvät siinä, että kaikki kuuntelevat toisiaan ja kaikilla on ajantasainen ymmärrys kokonaisuudesta, ja siitä mitä tapahtuu.**

Kaikissa yrityksissä **suunnittelijat pyrkivät istumaan kehittäjien kanssa samoissa tiloissa joko asiakkaan luona tai omalla toimistolla.** Täysiaikaisesti tämä ei aina onnistu suunnittelijoiden työstäessä muita projekteja. Työskentely asiakkaan tiloissa parantaa kommunikaatiota asiakkaan kanssa, jolloin vastauksia ja päätöksiä saadaan nopeammin, ja asiakasta on helpompi osallistaa. Asiakas myös näkee, miten tulosta syntyy. Läsnäolo asiakkaalla nähtiin mataloittavan kommunikaatiota ja lisäävän luottamusta tiimin ja asiakkaan välillä.

Sähköisiin kommunikaatiovälineisiin tukeudutaan, vaikka tiimi istuisi samassa tilassa, mutta etenkin silloin, kun etenkin asiakkaan tai tiimin jäsenen työskennellessä toisella paikkakunnalla. **Kommunikaatio heikkenee sähköisten välineiden myötä etenkin esimerkiksi silloin, kun asiakkaalla ja tiimillä on valmiiksi ajatusmaailmat etäällä**

toisistaan. Videopuhelu koetaan sähköisistä välineistä parhaimmaksi, koska sen aikana voidaan jakaa materiaalia ja saada keskustelulle visuaalista tukea. Heikoin kommunikointiväline on sähköposti.

6.5 Projektien ketterät käytännöt

Ketteryyttä on ennen kaikkea se, että **kehitystiimin ei tarvitse sitoutua koko projektin ajaksi yhteen menetelmään tai vain tiettyihin käytäntöihin**, vaan näiden suhteen on mahdollista olla joustava tilanteesta riippuen. Haastatteluissa tuotiin esille, että projekti voidaan aloittaa tiukalla Scrum-menetelmällä, ja projekti kunnolla käynnistyttyä siirrytään enemmän Leanin ja Kanbanin käytön suuntaan.

Haastatteluissa ilmeni, että ihmisten on hankala kuvata toimintamallejaan, jos niille ei ole valmiiksi määriteltyä struktuuria, johon käytännöt selkeästi pohjaavat. Yritykset ovat pyrkineet hahmottelemaan jonkinlaista omaa mallia, ellei yritys pyri toimimaan täysin Scrumin mukaisesti. Omat mallit toimivat viitekehyksinä, joiden mukaan pyritään toimimaan**. Omilla viitekehyksillä pyritään mahdollisimman Leaniksi, ja poistamaan turhat toiminnallisuudet ja tehtävät**, kuten yksityiskohtaiset rautalankamallit, tarpeettomiksi koetut palaverit ja kattavan dokumentaation kirjoittamisen. Palavereista jätetään väliin yleisimmin päiväpalaverit, katselmoinnit ja retrospektiivit. Vain kahdessa yrityksessä retrospektiivit kuuluvat vakituisiin käytäntöihin, tosin ei jokaisen sprintin jälkeen. Suunnittelijat kokevat, että retrospektiiveistä olisi hyötyä vähintään projektin loputtua, jotta prosessia pystytään parantamaan sisäisesti. Näitä ei kuitenkaan yleensä järjestetä.

6.5.1 Projektin aloitus

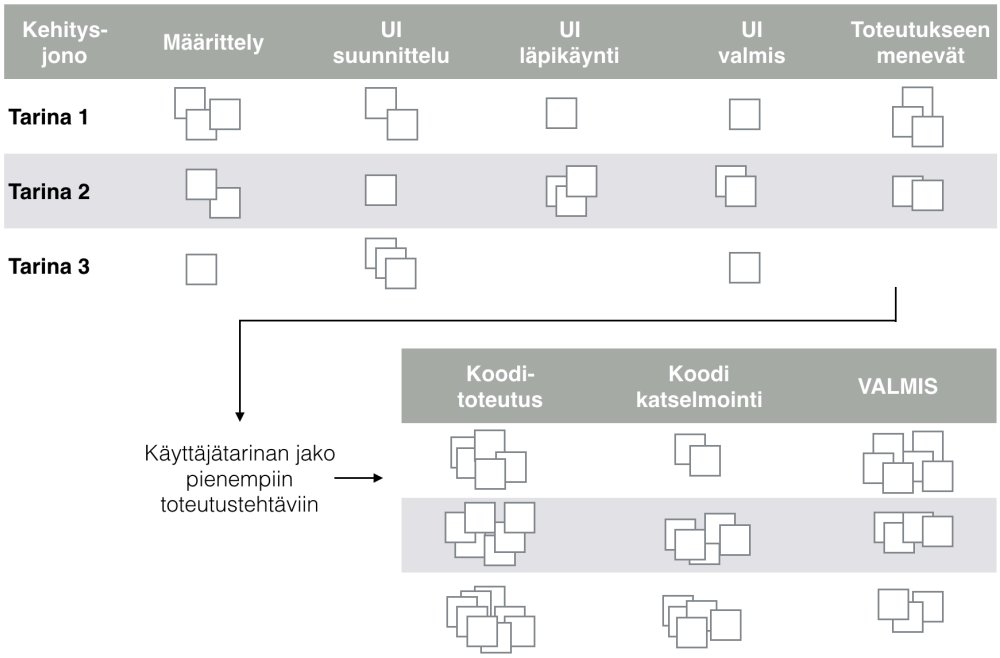
Asiakkaan kanssa pidettävä aloituspalaveri on projektin tärkein palaveri, ja siihen suurimmassa osassa yrityksistä osallistuu kaikki tiimin jäsenet. Asiakkaan kanssa pidettävää aloituspalaveria edeltää sisäinen aloituspalaveri, jossa käydään läpi, mitä asiakkaalle on myyty, mitä asiakkaalle on luvattu toimittaa ja mitkä ovat projektin tavoitteet myös sisäisesti. Asiakkaan luona fokusta tarkennetaan ja sovitaan yksityiskohdista, kuten toimintamalleista ja vastuualueista. **Asiakkaan osallistaminen projektiin aloitetaan jo ensimmäisessä palaverissa.** Palaverin ohessa voidaan pitää ensimmäinen yhteinen työpaja, jonka avulla voidaan määritellä tarkempia tavoitteita, odotuksia ja käyttäjäryhmiä. Työpajan tehtävillä saadaan asiakas tekemään ensimmäisiä päätöksiä tiimin johdolla.

6.5.2 Esiselvitys

Jokainen yritys pitää esiselvitysvaihetta tärkeimpänä suunnitteluvaiheena, ja harmittelee, jos sille on jätetty liian vähän aikaa. **Esiselvitystä pidetään koko projektin perustana.** Sen aikana **selvitetään esimerkiksi haastatteluilla ja työpajoilla asiakkaan tärkeimmät liiketoiminnalliset tavoitteet, loppukäyttäjien vaatimukset ja odotukset, sekä käyttäjänavigaatiot.** Toisinaan asiakas toimittaa valmiiksi tehdyn, satojen sivujen mittaisen esiselvityksen, ja vaatii käyttämään sitä. Valmiiksi toimitetun esiselvitysdokumentin laadusta ei ole aina takeita, sillä ne on voitu tehdä kehityksen tai markkinoinnin näkökulmasta, eikä loppukäyttäjän näkökulmasta, mikä on esiselvityksen varsinainen tavoite. Tällaisessa tilanteessa yritykset ovat oppineet vakuuttamaan asiakkaan siitä, että **itse tehtyä esiselvitystä ei korvaa mikään**. Viimeistään siinä vaiheessa, kun asiakkaalta on kysytty tarpeeksi monta kertaa ”*Miksi?*”, huomaa se itsekin, että myös siltä itseltään puuttuu todellinen ymmärrys asioista. Esiselvityksen perusteella luodaan ylätason konseptiluonnos tai -luonnoksia yhdessä asiakkaan ja käyttäjien kanssa. **Eräs haastateltava painotti konseptin tärkeyttä, sillä jos se määritellään väärin, lähtee projekti heti alussa väärään suuntaan.** Konseptiin sisältyy vision hahmottamista, toiminnallisuuksia, käyttöliittymän ja visuaalisen ilmeen ja mahdollisesti kevyttä prototyypin tekoa. Prototyyppejä saatetaan testata ja validoida loppukäyttäjillä, jos sille on aikaa. Esiselvitysmateriaalia käytetään vaatimusten määrittelyyn ja persoonien luontiin, joiden perusteella kootaan käyttötapaukset kehitysjonolle. Esiselvitystehtävät painottuvat vahvasti suunnittelijoille. Ainoastaan yhdessä yrityksessä on säännönmukaisesti osallistettu myös kehittäjät esiselvitystehtäviin. Kyseisessä yrityksessä koko tiimi on yhdessä vastuussa kokonaisuudesta, eikä tiimissä haluta olevan rooleja, vaan kompetensseja.

6.5.3 Kehitysjonon muodostaminen

Ennen iteraatioiden aloittamista luodaan kehitysjono käyttötapauksista. **Kehitysjonon kokoamisesta ovat yleensä vastuussa asiakas ja kehitystiimi yhdessä, mutta sen priorisointi mielellään jätetään asiakkaan vastuulle.** Asiakasta autetaan, jos ohjelmistokehitys tai ketterät menetelmät eivät ole tälle tuttuja. Viisi yritystä kuudesta käyttää tähän Kanban-taulua, jossa on koko tiimin tehtävät. Yrityksessä, jossa suunnittelijat ovat tiimistä erillään oleva osa, pitävät yllä omaa Kanban-taulua, joka on vain suunnittelijoiden nähtävillä, koska suunnittelijat eivät koe, että heidän tehtäviensä on lupa sotkea kehittäjien omaa Kanban-taulua. Kanban-tauluun voidaan tehdä omat sarakkeensa suunnittelijoiden tehtäville, jos se koetaan tarpeelliseksi, kuten alla olevassa kuvassa 11 on tehty. Joka tapauksessa **Kanban-taulun muoto luodaan tiimin omia tarpeita silmällä pitäen**. Tiimi saattaa pitää erillisiä sarakkeita Kanban-taululla villeille ideoille ja asioille, joita tuotteeseen olisi kiva lisätä. Tämä on yksi tapa tehdä käyttötapauksille karkeaa priorisointia.



Kuva 11 *Suunnittelu- ja kehitystyön erilliset Kanban-taulut*

6.5.4 Suunnitteluprosessi

Haastatteluissa kävi selväksi, ettei suunnitteluvaiheelle ole vakioitua prosessia, mutta kaikki suunnittelijat pyrkivät tekemään työnsä mahdollisimman kevyesti, koska työhön sisältyy paljon erilaisia tehtäviä. **Suunnitteluprosessin vakioimiselle ei koeta tarvetta,** koska projektien tarpeita tai tavoitteita ei voida vakioida. **Suunnittelumenetelmien valinta pohjautuu pääasiassa suunnittelijan kokemukseen, intuitioon ja projektin tarpeisiin.** Käyttäjäkeskeiseen suunnitteluun kuuluvia tehtäviä on koottu edellä esitettyyn taulukkoon 4. **Perustana suunnittelutyölle ovat persoonien, käyttäjätarinoiden, käyttötapausten ja skenaarioiden määrittäminen huolellisesti.** Kun nämä on tehty kunnolla, kommunikoitu koko tiimille ja saavutettu yhtenäinen ymmärrys visiosta ja isosta kuvasta, lopuilla tehtävillä enää varmistetaan onnistunut toimitus. Pienin mahdollinen suunnittelu toteutustyön aloittamiseksi voi olla pelkästään suunnittelijan sanallinen kuvaus toiminnallisuudesta etenkin. **Ketterintä olisikin vain piirtää ideat paperille tai tussitaululle kehittäjien kanssa.** Isommissakaan projekteissa erillisiä rautalankoja tai prototyyppejä ei ole aina tarvetta luoda, koska käyttöliittymäperiaatteet ja visuaalinen ulkonäkö seuraavat aiemmin määriteltyä tyyliä.

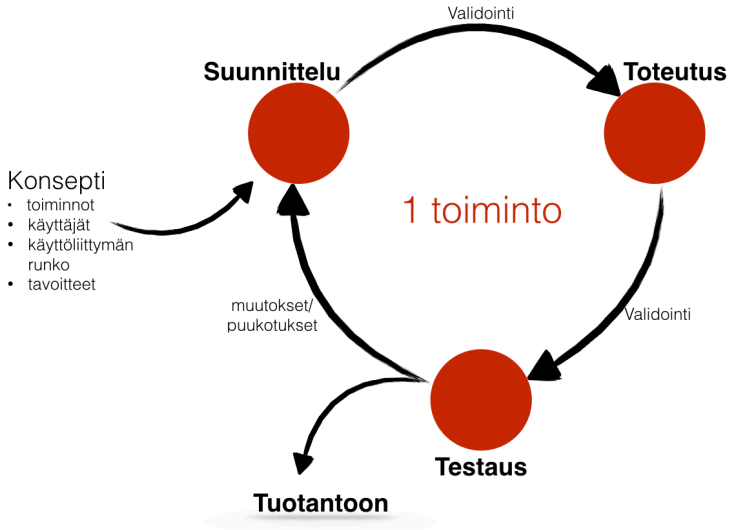
6.5.5 Suunnitteluratkaisujen iteroiminen

Kaikissa yrityksissä **suunnittelua pyritään tekemään yksi tai kaksi iteraatiota kehitystä edellä**. Suunnittelijat kokevat toisinaan sekavaksi tehtäviensä kokonaisuuden

hahmottamisen, kun pitäisi olla ymmärrys, mitä seuraavaksi tehdään ja työstää sitä sekä samalla seurata, mitä toteutuksessa tapahtuu ja tukea heitä. Lisäksi samanaikaisesti olisi hyvä tehdä testausta. **Iteraation pituus koetaan lyhyeksi näiden kaikkien tehtävien tekemiseen. Yhden viikon iteraatiota pidettiin silti parhaana**, koska iterointi nähdään silloin olevan tehokkainta, ja muutokset saadaan mukaan tuotteeseen nopeasti. Projektin alussa voidaan suunnittelulle antaa hieman enemmän aikaa luoda projektista ja sen toiminnallisuuksista iso kuva, joka ohjaa muuta suunnittelua. Suunnittelijoiden tehtävät ovat pääsääntöisesti isompia kokonaisuuksia kuin kehittäjien tehtävät. Suunnittelijat käsittelevät kokonaisuuksia käyttäjätarinoiden, skenaarioiden tai käyttäjäeeposten tasolla, jotka kehittäjät paloittelevat pienempiin osiin. Suunnittelutehtävien koot vaihtelevat paljon tiimeittäin ja projekteittain, eikä yhtä tehtävää välttämättä saada suunniteltua yhden sprintin aikana, jolloin on vaarana, että tehtävää siirretään eteenpäin, ja myöhemmin tehtävät kasaantuvat ja muodostavat projektille pullonkaulan. Haastatteluissa tuli esille, että **kehitystä edeltävä suunnitteluiteraatio tarkoittaa suunnittelun viimeistelyä** tuolloin, eikä niinkään toiminnallisuuden koko suunnittelua.

Yrityksissä on vaihtelua, miten suunnittelijat ja kehittäjät työskentelivät yhdessä iteraatioiden aikana. Yritys, joka roolien sijaan panostaa tiimeissä kompetensseihin, tekee haastattelujen perusteella eniten pari- ja ryhmäsuunnittelua, johon osallistuu sekä suunnittelu- että kehityskompetenssiosaajia. Toisessa ääripäässä on yritys, jossa suunnittelijat kokevat olevansa ulkopuolinen osa kehitystiimiä, jotka helposti myös ohitetaan suunnitteluratkaisuissa. Puolivälissä hajontaa on yritys, jossa suunnittelija toimii aktiivisesti myös kehittäjänä. **Yhteissuunnittelu ja kommunikointi tiimin kesken kulkevat käsikädessä.** Kun tiimi tuntee kuuluvansa yhteen ja olevansa yhdessä vastuussa lopputulemasta, myös kommunikaatio on luontevaa ja aktiivista. Yhteissuunnittelu helpottaa dokumentointitaakkaa, eikä kaikista toiminnallisuuksista tarvitse luoda tarkkoja rautalankamalleja, vaan käyttöliittymiä voidaan hahmotella tussitaululle tai paperilapuille. Toisinaan pelkällä keskustelulla päästään eteenpäin.

Iterointia voidaan tehdä monessa tasossa. Suunnitteluratkaisuja itsessään iteroidaan, samoin kuin toteutusta, mutta myös näiden välillä tehdään iteraatiota. Toisinaan testauksessa huomataan, ettei jokin toiminnallisuus toimi, jolloin se palaa toteutuksesta takaisin suunnittelupöydälle. **Toiminnallisuuden prioriteetista riippuen se otetaan joko heti työn alle, tai sijoitetaan priorisoidulle kehitysjonolle.** Alla olevassa kuvassa 12 on esitetty yhden haastatellun yrityksen piirtämä ”iteraatiomylly” yhden toiminnallisuuden osalta. Kaikki toiminnallisuudet eivät päädy tuotantoon, vaan jotkut siirretään suunnittelusta suoraan roskakoriin tai jatkokehitysideoihin.



Kuva 12 *Konseptin toiminnallisuudet suunnitellaan, toteutetaan ja testaan ennen tuotantoon viemistä*

Kokemukset suunnittelun ja kehitystyön integraatiosta ovat vaihtelevat. Toiset suunnittelijat kokevat, että toteutus ei halua olla iteratiivinen, ja pistävät vastaan muutoksille. Tällöin tiimin dynamiikassa todettiin olevan vikaa. Suunnittelijat kokevat, ettei heidän työtään arvosteta, ja heitä syyllistetään muutoksista, jos kehittäjät vastustavat muutoksia. Toisilla taas on hyviä kokemuksia integraatiosta. Kyseisissä yrityksissä panostetaan koko tiimin, suunnittelijoiden ja kehittäjien, keskinäiseen yhteistyöhön kompetenssien väleillä. **Tiivis yhteistyö säilyy, kun iteraatiot pidetään lyhyinä, ratkaisuista annetaan palautetta sekä suunnittelijoille että kehittäjille**, joten koko tiimi pääsee vaikuttamaan lopputuloksen. **Päätöksenteko nopeutuu lyhyissä iteraatioissa.** Toiminnallisuuksia lisätään tuotteelle vähitellen, kun edelliset toiminnallisuudet toimivat oikein.

6.5.6 Käyttäjäkeskeisen suunnittelun hallinta iteratiivisessa prosessissa

Iteratiivisuus tuo haastetta aikataulullisesti, koska suunnittelun pitäisi olla koko ajan toteutuksen edellä. **Työmääräresursoinnissa tulisi huomioida vaihtelevuus.** Projektin alussa pyöriä laitetaan vasta pyörimään ja suunnittelijoilla on työtä paljon. Kun projektin käytännöt ja iso kuva ovat kaikkien hallinnassa, suunnittelu helpottuu**. Työmäärän hallinnassa auttaa ymmärrys, mitä arvoa projektissa ollaan tuottamassa.** Tuote ei välttämättä näytä hyvältä, vaikka se tuottaa lisäarvoa**.** Tosin esteettinen tuote on helpompi myydä. Jos ylimääräisiä resursseja on saatavilla, voidaan tuotteen esteettisyyttä parantaa. Kyse on ennen kaikkea arvon tuottamisesta asiakkaalle ja käyttäjille.

Suunnittelijoiden kiire on yleisin syy pullonkaulojen syntyyn. Vaara niiden syntymiseen tulee, kun asiakas ei hyväksykään ehdotettua ratkaisua tai yllättäen vaatii muutoksia, jolloin suunnittelija joutuu siirtämään muita töitään eteenpäin. Tästä syntyy kerrannaisvaikutuksia myös kehitykselle. Pullonkaulan synnyttyä toteutus joutuu priorisoimaan omia tehtäviään uudelleen. **Suunnittelijoiden kiirettä lisää usean projektin samanaikainen työstäminen**. Tehtävien kasautuminen yhdessä projektissa vaikuttaa muihin projekteihin. **Pullonkauloja voidaan ehkäistä resuroimalla tarpeeksi aikaa esiselvitykseen ja ison kuvan varmistumiseen.** Tilanteita helpottaa myös front end -kehittäjien kokemus, jolloin he voivat tehdä suunnittelupäätöksiä itsenäisesti.

Ylipäätään **suunnittelijoille tulisi allokoida tarpeeksi aikaa projektille**, ja resursoinnissa pitää hyväksyä, ettei suunnittelijoiden tarvitse olla jatkuvasti työn touhussa, vaan **työmäärässä on vaihtelua.** Suunnittelijat toivovat oman yrityksen sisäisessä resursoinnissa järkevyyttä siihen, moneenko projektiin suunnittelijan tulee keskittyä kerralla, ja kyseenalaistavat johdon tarpeen resursoida suunnittelijat aina projekteihin 100 % allokaatiolla, jolloin aikataulullisesti ei ole joustavuutta muutoksille ja asiakkaan päätöksenteon viivästyksille.

6.5.7 Käyttäjätestaus iteratiivisessa prosessissa

Käyttäjätestaus koetaan tarpeellisena ja muutama suunnittelija toivoisi niitä olevan enemmän. Käyttäjätestaus on kuitenkin jätetty usein pois, koska **se koetaan aikaa vieväksi ja vaivalloiseksi saatuihin tuloksiin nähden.** Käyttäjätestauksen haasteena on sovittaa se nopeatempoiseen suunnittelu- ja kehitysrytmiin. Yksi yritys kommentoi, että **testauksella saadaan helposti selville opittavuusongelmat, mutta niiden avulla ei nähdä, jos kokonaiskonseptissa tai tehokkuudessa on ongelma**, jotka ovat kriittisiä ongelmia lopputuotteelle. Kyseiset haastateltavat kokevat, että samaan lopputulokseen päästään läpikäymällä tuotos yhdessä asiakkaan kanssa. Käyttäjätestausta on näin ollen korvattu sisäisellä asiantuntija-arviolla, läpikäynnillä asiakkaan kanssa tai suunnittelijan tekemällä toteutuksen katselmoinnilla ennen sen julkaisua. Tässä todettiin olevan eettinen ongelma, sillä suunnittelija on aina kallellaan omien ratkaisujensa suuntaan.

Projektin loppuvaiheessa tehtävä käyttäjätestaus nähdään tarpeettomaksi, koska tuolloin tuotteeseen ei voida tehdä enää muutoksia. **Käyttäjätestaus on tärkeintä tehdä konseptivaiheessa prototyypille**, jotta varmistutaan tuotteen tai palvelun oikeasta suunnasta, ja ratkaiseeko se käyttäjien ongelman.

6.5.8 Muutoshallinta vastaan iteratiivisuus

Iteratiivisuuden idea on, että muutoksia voidaan helposti tuoda mukaan tuotteeseen, ja muutostarpeisiin kyetään reagoimaan. **Haasteena on, että iteraation sisältö saatetaan**

nähdä lukkoon lyötynä, vaikka iteraation aikana olisi aikaa hioa ratkaisua. Haastatteluissa muutoksia ei koettu ongelmaksi suunnittelijoiden osalta. On kuitenkin olemassa muutoksia, jotka koetaan epämiellyttäviksi. Tällaisia muutokset johtuvat asiakkaan toiminnasta, eivätkä niinkään kohdistu kehitettävään tuotteeseen. Alle on listattu esimerkkejä tämänkaltaisista muutoksista:

- Kehitystiimit kokevat turhautumista, kun asiakkaan jo hyväksymä ja mahdollisesti toteutettu toiminnallisuus päätetään repiä auki ilman kunnollisia perusteluja. Muutokset vaativat aina hyvät perustelut, koska ne vaikuttavat muihin toiminnallisuuksiin, ja siirtävät muita töitä myöhemmäksi.
- Toisinaan asiakas päättää jyrätä jonkin asian läpi ilman, että yhdessä mietitään, onko toiminnallisuus järkevä. Tällaiset vaatimukset tulevat yleensä liiketoiminnan puolelta, ovat poliittisia linjauksia tai joku organisaation hierarkiassa korkealla oleva taho on kesken projektin herännyt, ja ilmoittaa mielipiteensä. Suunnittelijoiden oletetaan olevan mieliksi asiakasyritysten pomoille oman työnsä laadun ja käyttäjien näkökulman kustannuksella.
- Asiakkaiden on vaikea hyväksyä ajatusta, ettei heidän back end taivu tulevaisuuden vaatimuksiin, ja he joutuvat investoimaan isolla rahalla tämän tyyppisiin muutoksiin. Asiakas monesti ennemmin hyväksyy huonon tuotteen kuin investoi nykyaikaiseen järjestelmään.

Asiakkaan tulee aina hyväksyä isommat muutokset, jotka liittyvät esimerkiksi tietosisältöön, uusiin toiminnallisuuksiin, koska asiakkaalla on lopullinen päätösvastuu. **Jos asiakkaan läpikäynnistä tai muuten saadusta palautteesta aiheutuu muutoksia, suunnittelijan tulisi hyväksyä ne ennen toteuttamista.** Muutokset kirjataan tuotteen kehitysjonolle priorisoitavaksi. Muutostarpeita luokitellaan myös tulevien versioiden kehitysjonoihin, ja niin kutsuttuun Toiveiden tynnyriin.

6.5.9 Dokumentointi

Kaikki yritykset ovat luopuneet suunnitteluvaiheen erillisestä dokumentoinnista. **Dokumenteiksi riittää tussitauluista otetut valokuvat, skannatut paperiraapustukset, rautalankakuvat, ja grafiikkakuvat, sekä muistiot suunnittelupalavereista ja mahdollisesti arkistoitavat käyttötapaukset.** Asiakkaat kysyvät dokumentaation perään, mutta ymmärtävät sen vievän suunnittelijoilta paljon resursseja, joka on pois suunnittelusta. Dokumentaatiolla ei ole merkitystä sen jälkeen, kun tuote on julkaistu. Jos tuotetta myöhemmin kehitetään ja toimittaja vaihtuu, uudet kehittäjät joutuvat joka tapauksessa käymään koodin läpi, ja suunnittelijat selaamaan ja arvioimaan käyttöliittymän uudelleen. Dokumentaatio on myös huono kommunikointiväline.

6.6 Asiakkaiden mukanaolo projektissa

Asiakkaan kokemus ohjelmistoprojekteista ja ketteristä menetelmistä vaihtelee paljon asiakkaan toimialan mukaan. **Useimmiten asiakasta joudutaan jollakin tasolla kouluttamaan ketterään ajatteluun.** On hyvä, jos asiakas myöntää jo alussa, ettei ymmärrä menetelmää, jolloin koulutukseen voidaan varautua etukäteen. Joskus projektiin osallistuvat asiakkaan edustajat ymmärtävät ja omaksuvat idean loistavasti, mutta heidän takana oleva johtoryhmä suhtautuu toimintatapaan jähmeästi, koska ovat tottuneet klassiseen vesiputousmalliin. Tällöin menetelmän valinnasta saatetaan joutua vääntämään kättä. **Joidenkin asiakkaiden kanssa ongelmana on iteratiivisen toimintatavan istuminen asiakkaan omiin prosesseihin**, kuten hyväksymisprosessiin. Toisaalta on asiakkaita, jotka eivät itse suostu olemaan ketteriä, mutta odottavat kehitystiimin olevan. Käytännössä tämä tarkoittaa, että kehitystiimin tulee reagoida muutoksiin heti, mutta kehitystiimi ei itse saa vastauksia välttämättä ajoissa. **Yleisesti asiakkaat ovat ketterään toimintatapaan tyytyväisiä**, koska saavat nopeasti käsiinsä jotakin valmista. Kun asiakkaat ymmärtävät ketteryyden ja kokevat pääsevänsä aidosti vaikuttamaan lopputulokseen, he ovat entistä onnellisimpia, jopa kankeammissa yrityksissä. Monilta yrityksiltä on jälkikäteen tullut kiitosta ja palautetta, että tämä on järkevä tapa toimia.

On ensisijaisen tärkeää projektin onnistumisen kannalta, että **asiakas sitoutuu projektiin. Suurin osa haastatelluista haluaa tästä syystä asiakkaan ottavan tuoteomistajan roolin, ja asiakkaan allokoivan tuoteomistajalle tarpeeksi aikaa roolia varten.** Tällä varmistutaan siitä, että asiakkaan puolella on projektista vastaava henkilö, ja joku, jolle avoimet kysymykset osoitetaan selvitettäväksi. Haasteena saattaa olla asiakkaan kokemattomuus ketteristä menetelmistä tai ohjelmistokehityksestä, jolloin prosessin eteenpäin vieminen on luonnostaan pääasiassa toimittavalle yritykselle. Tästä huolimatta **asiakkaan tulisi osallistua käyttötapausten priorisointiin, päiväpalavereihin ja päätöksentekoon, tai asioiden eteneminen hankaloituu merkittävästi**. Asiakkaan sitoutumista ja osallistumista yleisimmin haittaa se, että projektiin liittyvät tehtävät tehdään muiden töiden ohella, eikä projektille ole allokoitu osuutta työajasta.

Asiakkaan tulisi olla proaktiivinen ja kiinnostunut siitä, mitä tehdään. **Aktiivinen, päätöksentekoon valmis asiakas, jolla on oikeaa halua ja innostusta tehdä projektia, on optimaalinen.** Projektiin ei tulisi osallistaa asiakkaalta henkilöitä ainoastaan esimiehen määräyksestä, vaan heidän tulee olla motivoituneita yhteistyöhön. Asiakkaan toivotaan valmistautuvan palavereihin ja läpikäyntitilaisuuksiin riittävällä tavalla, jotta tapaamisessa voidaan keskittyä olennaiseen. **Hyvä asiakas antaa myös jatkuvaa palautetta, ja ymmärtää, ettei itse välttämättä ole tuotteen loppukäyttäjä.** On hyvin yleistä, ettei asiakas ymmärrä oman roolinsa ja loppukäyttäjän eroa. Asiakkaan kanssa yhteistyö sujuu mainiosti, jos asiakas luottaa suunnittelijoihin ja kehittäjiin alan ammattilaisina. Toisinaan käy niin, että asiakas on liian kiinnostunut projektista, ja ottaa sen

liian omakseen haluamalla osallistua kaikkeen, mitä kehitystiimi tekee. Tämä saattaa häiritä kehitystiimin työtä ja projektissa etenemistä. Kehitystiimi tarvitsee myös työrauhan.

"*Parhaat projektit ovat sellaisia, joissa ei oikeastaan tiedä,*
kuka on asiakkaalla töissä ja kuka meillä. Tehdään siis yhdessä tätä projektia. "

7. VERKKOAINEISTOANALYYSIN TARKASTELU

Ohjelmistoritykset ymmärtävät hyvin valitsemiensa menetelmienn perusteet, toiminta-periaatteet ja prosessit. Koska kaikki ketterät mallit ovat filosofioita ja viitekehyksiä, jokaisen yrityksen tulee itse löytää tapa soveltaa menetelmiä omiin toimintoihin ja prosesseihin. Soveltaminen onkin haaste yrityksille, koska sitä ei voida täysin kopioida keneltäkään toiselta, vaan se vaatii epäonnistumisia, oppimista ja jatkuvaa parantamista. Internetin kautta yritysten on helppo oppia muiden virheistä, jakaa omia kokemuksia, kysyä neuvoa ja keskustella asioista globaalisti. Internetissä on paljon suunnittelijoiden, kehittäjien, projektipäälliköiden ja asiakkaiden blogeja, artikkeleita, keskusteluja ja konferenssiesityksiä, joista saada lisää tietoa menetelmien soveltamisesta.

Tässä luvussa verkkoaineiston tulosten läpikäynnin lisäksi yhdistetään tuloksiin joitain haastatteluista esiin tulleita tuloksia. Artikkelit, blogit ja niistä löytyneet, alalla työskentelevien käymät keskustelut, sekä työyhteisössäni käymät keskustelut antavat perspektiiviä tämän hetken todellisiin käytäntöihin, niiden haasteisiin ja ratkaisuihin. Tärkeimmät tulokset on korostettu lihavoinnilla.

7.1 Suunnittelijan rooli ja tehtävät ketterässä tiimissä

Suunnittelijat eivät ole projektin ainoita henkilöitä, jotka vastaavat suunnittelutehtävistä, vaan myös muilla tiimin jäsenillä saattaa olla kompetenssia ja halua osallistua näihin. Suunnittelijoiden kompetenssiosaaminen voi myös ylettyä kehitystiimin tehtävien puolelle. **Tiimin jäsenille tulisikin sallia liikkumavara osallistua monipuolisesti projektin tehtäviin heidän motivaationsa mukaan.** Tämä liittyy vahvasti tiimin itseohjautuvuuteen, jota käsitellään tarkemmin kappaleessa 7.2.

Haastatteluissa todettiin, että ketterässä tiimissä on yhdestä kahteen suunnittelijaa. **Parisuunnittelu koettiinkin mieleisimmäksi tavaksi tehdä suunnittelua**, koska suunnittelijat kokivat hyötyvänsä siitä, että toinen suunnittelija haastaa heidän ratkaisujaan ja yhteistyöstä syntyy helpommin innovatiivisia ratkaisuja. Työkuorma helpottuu, kun toinen suunnittelija on mukana ideoimassa ja avustamassa. Suunnittelijan vierellä voi myös toimia front end -kehittäjä, jonka muutenkin tulisi validoida ja antaa palautetta suunnittelusta. Kun suunnittelijan parina toimii kehittäjä, vähentyy riski erottaa suunnittelijat kehitystiimin ulkoiseksi osaksi.

Haastatteluiden ja internet-lähteiden [26, 72, 91] perusteella **suunnittelijoiden tehtäviksi muodostuu ennen kaikkea esiselvitykseen, interaktio- ja käyttöliittymäsuun-**

nitteluun sekä käyttäjätestaukseen liittyvät tehtävät. Suunnittelijoiden työ nähdään hyvin itseohjautuvana, etenkin ketterissä ohjelmistoprojekteissa, jolloin suunnittelijat itse arvioivat, millä menetelmillä ja toimintatavoilla kyseiselle projektille tuotetaan paras mahdollinen tulos mahdollisimman tehokkaasti. [76] Lisäksi **suunnittelijoiden kompetenssialueet osaltaan määrittelevät projektissa käytetyt suunnittelumenetelmät ja suunnittelutehtävät**. Suunnittelijoiden osaamisalueeseen odotetaan kuitenkin kuuluvan tiettyjä toimintoja ja tehtäviä, jotka liittyvän muun muassa seuraan asioihin [26, 72, 91]:

- vastuu vision ja kokonaiskuvan hallinnasta,
- oikeiden käyttäjien osallistaminen,
- toimiminen linkkinä käyttäjien ja kehittäjien välillä,
- vaatimusten kerääminen eri sidosryhmiltä, niiden määrittely ja hallinta,
- käyttäjätarinoiden ja käyttötapausten luominen,
- kehitystiimin kouluttaminen ymmärtämään käyttäjätarpeita ja muuttaa kehittäjien roolia enemmän käyttäjien tukijoiksi,
- kehitysjonon hallinta ja priorisointi,
- testitapausten määrittely ja suunnittelu,
- käyttäjätestien järjestäminen,
- antaa takuu tuotteen laadusta ja
- palvelusuunnittelu.

Kuten listauksesta näkyy, **suunnittelijoiden työ on kokonaisvaltaista vaatimusten, ratkaisujen ja laadunvarmistuksen hallintaa**. [87] Tästä syystä suunnittelijaa ei saisi ottaa projektiin mukaan liian myöhään, koska sillä on haittaava vaikutus kokonaisvaltaisen käyttäjäkokemuksen suunnitteluun, joka lähtee liikkeelle jo visiosta ja strategiasta.

Suunnittelijoiden osaaminen on arvossaan määriteltäessä MVP-versiota ja sen toiminnallisuuksia, sillä kehittäjille ja tuoteomistajalle tämä voi olla hankalaa. **Käyttäjäkokemusammattilaisilla on kyky piirtää selkeät raamit tavoitteista, periaatteista ja visiosta, sekä auttaa päätöksissä**, kuten milloin jokin toiminnallisuus on hyvä julkaista. [91] Suunnittelijoiden on myös kyky kiinnittää huomiota enemmän viimeistelyyn kuin kehittäjillä ja tuoteomistajilla, antaa suunnittelijoille vastuun tarkastaa kehityksen tuotokset ja muistuttaa heitä laadukkaasta toteutuksesta ja loppukäyttäjille tuotettavan käyttäjäkokemuksen tärkeydestä. [87]

Suunnittelijalla saattaa olla useampia projekteja samanaikaisesti allokoituna, kuten haastatteluissa todettiin. Jos suunnittelija joutuu jakamaan huomionsa usean projektin välillä ja hoitamaan usean iteraation asioita samanaikaisesti, kehitysprosessista saattaa muodostua kaoottinen ja kankea. [74] Suunnittelijan voi olla hankala priorisoida suunnittelutehtäviä eri projektien välillä ja osallistua kaikkiin tarvittaviin palavereihin. Visi-

on ylläpitäminen heikkenee, jos suunnittelija ei ole mukana koko kehityskaaren ajan. [87, 81] Kehitystiimille saattaa myös hämärtyä, mikä on suunnittelija rooli projektissa, ja milloin suunnittelija aikoo osallistua projektin tekoon, jos samanaikaisia projekteja on useita. [87] Tällä on suora vaikutus tiimin sisäiseen kommunikaatioon, jota on käsitelty lisää kappaleissa 5.3.4 ja 6.3. **Kun suunnittelija osallistuu projektiin aktiivisesti ja seuraa keskustelua, tiimi ymmärtää, että suunnittelija on paikalla ja yrittää auttaa** [87].

Harris kertoo kirjoituksessaan [74] omakohtaisen ratkaisun, jossa jokaisen projektin Kanban-taulut yhdistetään yhdeksi suunnittelutehtävien osalta. Ennen jokaista iteraatiota tai kerran kuukaudessa yhdistelmätaulun käyttäjätarinat katsottiin läpi ja siistittiin yhdessä jokaisen projektin tuoteomistajan tai projektipäällikön kanssa. Tällöin tieto suunnittelijan kokonaistilanteesta välittyi yksittäisille tiimeille.

Suunnittelutyön sanottiin haastatteluissa toisinaan osoittautuvan projektin pullonkaulaksi erinäisistä syistä. **Suunnittelijalle tärkeä ominaisuus on hallita omaa ajankäyttöään, ja kyetä tekemään kompromisseja ja valintoja**, jotta projekti voi edetä. [87, 95] Yksi keino välttää suunnittelun muodostumista pullonkaulaksi on **määritellä projektin alussa suunnitteluperiaatteet selkeästi**, jolloin kehityksen ohjaaminen eteenpäin spontaanisti keskustelemalla helpottuu. [45]

7.2 Tiimin integroituminen yhdeksi

On tärkeää luoda monialainen tiimi [82, 95]. Vahva tiimi tuntee toisensa hyvin, osallistaa asiakasta säännöllisesti ja on omistautunut. Agile HR [76] ja Kiminki [78] huomauttavat, että motivoituneita tekijöitä ovat ne, jotka itse hakeutuvat kyseiseen projektiin ja tiimiin. **Motivoitunut ja itseohjautuva tiimi saa aikaan parhaat arkkitehtuurit, vaatimukset ja suunnitelmat.** [78] Käytännössä tämä on usein haaste, sillä projektien kannalta asiantuntijat eivät voi vaihtua jatkuvasti, joten tiimit pitää koota useimmiten vapaana olevista tai kohta vapautumassa olevista resursseista. Erityisen suuri haaste on pienillä yrityksillä saada kaikkiin projekteihin motivoituneet asiantuntijat. Yrityksillä on apuna erilaisia resursointiin liittyviä menetelmiä ja välineitä, jotka auttavat tiimin kokoamisessa.

On olemassa vastakkaisia mielipiteitä siitä, pitäisikö tiimissä jo projektin alussa päättää, kuka tekee mitäkin tehtäviä, ja kuka on mistäkin tehtävästä vastuussa. Muun muassa [81, 87] lähteen puoltavat ainakin jokseenkin roolien ja tehtävien etukäteen päättämistä, kun taas lähteet [76, 78] puoltavat toimenkuvien joustavuutta ja liikkumavaraa. Etukäteen määrittämistä perustellaan yhteisellä ymmärryksellä vastuualueista, kompetensseista ja taustoista. Näiden selkeys ei kuitenkaan ole riippuvainen siitä määritelläänkö roolit ja tehtävät etukäteen**. Jokaisen asiantuntijan kompetenssiosaaminen ja kokemus-**

tausta on hyvä kommunikoida projektin alussa kaikille, jolloin vastuualueet luonnollisesti ohjautuvat niiden mukaan. **Etukäteen päättäminen vastuualueista lokeroi tekijöitä rooleihin ja vähentää tiimin yhteisvastuuta kokonaisuudesta**, kun vastuu tehtävistä yksilöidään. Haastatteluissa yksi firma selkeästi ilmaisi, etteivät he kannata roolien ja tehtävien omistajuuden määrittämistä, koska näkevät paremmaksi sen, että tiimi on yhdessä vastuussa kokonaisuudesta, vaikka tehtävät jakautuvatkin luonnostaan kompetenssien mukaan. Joustavuus mahdollistaa motivoituneiden kehittäjien osallistumisen suunnittelutehtäviin ja päinvastoin, mikä taas tukee motivaatiota.

Itseohjautuvuus tarkoittaa vastuuta ja valtaa samassa paikassa [76]. Haastattelujen lisäksi monissa keskusteluissa ja internet-lähteissä [52, 76, 92] ollaan yhtä mieltä siitä, että tiimin tulee olla itseohjautuva. Van Weerdenburg [92] listaa tiimin alhaisen itsenäisyyden ja itseohjautuvuuden yhdeksi kehitystiimiä hajottavaksi tekijäksi. **Itseohjautuvuudella ja yhteisvastuulla varmistetaan yhteinen käsitys projektin kokonaiskuvasta, ja jaetaan vastuu kokonaisuudesta kollektiivisesti.** Itseohjautuvuus on kuitenkin käsite, joka pitäisi määritellä. Liiketoiminnallisesta näkökulmasta tiimin itseohjautuvuutta rajoittavat tiimille ja projektille asetetut lopulliset tavoitteet, välitavoitteet ja mittarit. Tiimillä tulee olla hyvä näkemys, mihin projektissa pitää päästä. Tätä ohjaa ensisijaisesti visio, joka muodostetaan projektin alussa. **Tiimin itseohjautuvuus näin ollen rajoittuu siis menetelmiin ja työkaluihin, joilla päästään tavoitteisiin ja toteutetaan visio.**

Yhteisvastuu ja itseohjautuvuus onnistuvat vain, jos suunnittelijat ovat täysin integroituneet osaksi kehitystiimiä, ja tiimi tekee työtä yhdessä. Tiimin dynamiikan tulisi olla sellainen, että tiimi haluaa työskennellä yhdessä. [90] Vaillinainen integraatio mitä todennäköisemmin aiheuttaa lisätöitä, kun kommunikaatio tiimin jäsenten välillä heikkenee. **Suunnittelijoiden integroituminen heikkenee, jos suunnittelijat työskentelevät useissa projektissa samanaikaisesti.** Tällaisessa tilanteessa olevat suunnittelijat ilmoittavat todennäköisimmin ajanhallintaan ja tiimin integroitumiseen liittyvistä ongelmista [87].

7.3 Tiimin kommunikaatio

Haastattelujen perusteella suunnittelutyön onnistunut integroiminen ketteriin prosesseihin on paljon kiinni kommunikaatiosta. Sen tärkeys ja siinä epäonnistumisen helppous ymmärretään. Hyvä kommunikaatio on välttämätöntä missä tahansa organisaatiossa riippumatta toimialasta tai prosessimenetelmästä.

Sekä haastatteluissa että artikkeleissa [26, 78] todetaan, että **tehokkain ja toimivin tapa tiedon välittämiseksi sidosryhmille, kehitystiimille ja tiimin jäsenten kesken on kasvokkain käytävä keskustelu**. Kommunikaation tulee olla jatkuvaa ja aktiivista. Tämän edellytyksenä on, että koko kehitystiimi istuu samassa tilassa toistensa vieressä

koko prosessin ajan [84, 87, 90], sillä mikään ei ole niin tehokasta kommunikoida spontaanisti kollegan kanssa kasvotusten, näyttää mitä on saatu aikaan, vaihtaa ajatuksia ja samalla ymmärtää kasvonilmeitä ja kehonkieltä [87]. Non-verbaalisen viestinnän osuus kommunikoinnista on merkittävästi suurempi kuin verbaalisen, joten kommunikointi on vajavaista, jos sitä käydään pääasiassa sähköisten välineiden kautta. Yhdessä työskentely johtaa luovempaan suunnitteluun ja ideointiin, koska kommunikaatiossa on mukana non-verbaalinen ilmaisu ja kommunikaatio on dynaamista. Tämä johtaa esteettisempään ja yhdenmukaisempaan tuotteeseen [69, 87]. **Jos suunnittelija ei ole saatavilla suunnittelukysymysten noustessa esiin, ne muodostavat projektille joko pullonkauloja, tai ne ohitetaan ja käyttäjäkokemusnäkökulma jää käsittelemättä.** Suunnittelijan läsnäolo helpottaa kysymysten esiin nostamista ja auttaa pitämään käyttäjien tarpeet fokuksessa [91].

Ketterät prosessit on rakennettu nopealle palautteelle, jota myös suunnittelija ja kehittäjä antavat toisilleen. **Palautteen antaminen helpottuu ja nopeutuu, kun tiimi istuu samassa tilassa ja tarkkailevat jatkuvasti toistensa aikaansaannoksia.** Tuotteen laatu paranee ja projektista tulee kustannustehokkaampaa, kun toinen silmäpari käy läpi tuotoksia ja löytää puutteita. Läheinen työskentely ja palautteenanto lisäävät suunnittelijoiden ymmärrystä kehityksestä, ja mitä on mahdollista toteuttaa, sekä kehityksen ymmärrystä suunnittelusta [69, 87, 91].

Käytännössä on tilanteita, joissa koko tiimin ei kannata tai se ei voi istua samassa paikassa. Tällaisia ovat esimerkiksi konseptointiprojektit, joissa suunnittelijoilla on suuri rooli, ja kehittäjä tekee kevyen prototyypin, tai enemmän kehitykseen painottuva projekti, jolloin suunnittelijan jatkuva mukana olo ei tuo tuotteelle lisäarvoa. Tästä syystä kehitystiimille on hyvä tarjota sekä **synkronoituja että asynkronoituja kommunikointikanavia** [87].

7.4 Suunnittelu- ja kehitysprosessi

Ketteryyden näkökulmasta on oikeasoppimista olla suunnittelematta isosti etukäteen, koska ymmärrys lisääntyy kehityksen edetessä. Suunnittelussa taas halutaan perinteisesti tietää kaikki merkitykselliset asiat etukäteen, jotta voidaan alusta asti varmistaa hyvä käyttäjäkokemus todellisessa käyttöympäristössä. Ketterä ohjelmistokehitys, johon on integroitu suunnittelutyö, on malli, josta ei voi piirtää selkeää karttaa edes organisaatiotasolla. **Jokainen projekti muodostaa prosessin itselleen sopivaksi.** Tämä ei kuitenkaan tarkoita, etteikö prosessi olisi tärkeä. Sekä suunnittelu että ketterä ohjelmistokehitys perustuvat iterointiin, toistuvaan palautteen keräämiseen ja käyttäjien tarpeisiin. Prosessin määrittäminen projektille sopivaksi käsittääkin ennen kaikkea työkaluja, toimintamalleja, tehtäviä, ja milloin suunnittelutyö tehdään ja miten kattavana.

Projektin alussa luodaan selkeä visio projektille ja tuotteen nautinnolliselle käyttäjäkokemukselle. Suunnittelijoilla on näiden määrittämiseen olemassa useita erilaisia menetelmiä kuten storyboard tai käyttäjätarina. Etupainotteinen vision luominen on erinomainen resurssi-investointi, sillä sen avulla vältetään massiivista uudelleen suunnittelua ja kehitystä [91]. Tästä syystä on tärkeää antaa vision luomiseen ja iterointiin tarpeeksi aikaa. Visio luodaan nollaiteraatiossa, jossa tehdään myös käyttäjätutkimusta. Käyttäjätutkimuksella voidaan varmistua oikeasta visiosta sen lisäksi, että se toimii persoonien ja käyttäjätarinoiden perustana. **Selkeät ja ymmärrettävät visio ja käyttötapaukset vähentävät turhautumista ja mahdollistavat tehokkaan työskentelyn** aikataulun ollessa tiukka [45, 90, 91]. Visio tulee pitää tarpeeksi korkealla tasolla, eikä siihen tule etukäteen listata jokaista projektin aikana eteen tulevaa käännekohtaa, jotta visio ei ohjaile liikaa tuotteen lopputilaa. Visiolla määritellään tuotteelle suunta, mutta yksityiskohtien annetaan hahmottua myöhemmin. Visio voidaan kommunikoida monella eri tavalla, kuten asiakkaan etenemissuunnitelmana, tarkastelemalla arvolupausta tai luomalla palvelukuvaus. [45] **Hyvin määritelty visio auttaa priorisoimaan käyttäjätarinoita** [81] **projektin seuraavissa vaiheissa.**

Käyttäjätutkimus tehdään nollaiteraatiossa, jonka aikana ei vielä tehdä kehitystyötä. **Käyttäjätutkimusta, kuten suunnitteluakin, tehdään vain tarpeellinen määrä**, joka riippuu aina projektista, eikä yhtään enempää. [76] **Koko tiimin tulisi osallistua käyttäjätutkimusvaiheen tehtäviin**, myös kehittäjien [91]. Näin varmistutaan siitä, että **koko tiimille muodostuu yhteinen ymmärrys ratkaistavasta ongelmasta ja ratkaisujen muodostuminen pohjautuu yhteiseen näkemykseen** käyttäjän tarpeista, käyttöympäristöstä ja vaatimuksista [81], eikä tätä kaikkea tarvitse tarkasti dokumentoida, sillä kehittäjät eivät kuitenkaan lukisi dokumentaatiota [90]. Koko tiimin ymmärtää myös liiketoiminnallinen ongelma, jota ollaan ratkaisemassa. Näiden pohjalta kehitystiimi yhdessä muodostaa oletuksia ja hypoteeseja, joita voidaan tarvittaessa testata jo nollaiteraation aikana kevyillä prototyyppiluonnoksilla ja haastatteluilla. [82, 90] **Kehittäjien osallistuminen suunnittelutehtäviin lisää luottamusta ja kunnioitusta suunnittelijoiden tekemää työtä kohtaan**, eikä suunnittelijoiden tarvitse perustella päätöksiään perusteellisesti, kun koko tiimi on ollut mukana esiselvityksestä lähtien. [87] Tuoteomistajien ja projektipäälliköiden näkökulmasta projektin alussa on hyvä varata aikaa nollaiteraatiolle vision luomisen lisäksi siksi, että tehty esiselvitys vähentää hukan määrää projektin myöhemmissä vaiheissa, kun testauksessa huomattavien ongelmien määrä vähenee ja uudelleensuunnittelua tarvitse tehdä.

Kun toteutusiteraatiot käynnistyvät, suunnittelutyötä voidaan tehdä yhtä, tai isommissa projekteissa kahta, iteraatiota kehitystä edellä. [81, 83, 90] Toisinaan kehitystiimit kokevat, että toimintatapa ei sovellu heille, ja siksi ajatus ketterästä kehityksestä muodostuu negatiiviseksi. Haastatteluissa suunnittelijat kommentoivat ajan puutetta, joka johtuu muun muassa monista saman aikaisista tehtävistä ja rooleista. Six [91] huomauttaa,

että tämän **toimintatavan onnistuminen on ennen kaikkea kiinni ihmisistä**. Nielsen [83] tarkentaa, että kyse on ihmisten asenteesta. **On todennäköistä, että suunnittelijat eivät kykene tekemään suunnittelua yhtä iteraatiota kehitystä edellä heti alussa,** koska kommunikaatiotavat, tarvittavan informaation jakaminen oikeassa muodossa ja vastuiden ja dynaaminen yhteistyö vaativat harjoittelua. Saattaa mennä muutama iteraatio ennen kuin kehitystiimille muodostuu oma tapa toimia, ja toimintatapa asettuu.

On hyvä huomioida, että kehitystä edeltävän iteraation ei tarvitse olla pelkkää suunnittelutyötä, sillä **front end -kehitys on osa suunnittelua**. Front end -kehityksen aloittamista suunnittelutyön ohella tukee se, että kehitystyö tulisi ketterien periaatteiden mukaan aloittaa mahdollisimman aikaisin. [91] Kun front end -kehittäjä työskentelee suunnittelijan rinnalla, jakautuu vastuu käyttäjäkokemuksen omistajuudesta laajemmin. Fotolescu [72] sanoo, ettei kaikkea suunnittelua kannata tehdä etukäteen, sillä se aiheuttaa lähinnä kysymyksiä siitä, mitä pitäisi tehdä ja mitä on jo tehty. Hänen mukaansa etukäteen suunnittelu on kankea lähestymistapa omaksua, ja aiheuttaa hämmennystä ja turhautumista. Lisäksi suunnittelijoiden ja kehittäjien välillä ei synny jaettua omistajuutta käyttäjäkokemuksesta.

Ramsay [86] muistuttaa, että **suunnittelijat eivät suunnittele yhtä iteraatiota kehitystä edellä, vaan koko tiimi suunnittelee yhdessä**. Suunnitteluideat tulisi koota tiiminä, jolloin rakennetaan yhteinen ymmärrys, mitä ollaan tekemässä. Tiimi yhdessä päättää, miten ongelmat ratkaistaan, jotta varmistutaan kyvystä tuottaa arvoa. Tästä syystä suunnittelijoiden pitää olla integroituna kehitystiimiin. [82, 87, 90] **Kun tiimi yhdessä ratkaisee ongelmia, jakautuu suunnitteluratkaisujen omistajuus kaikkien kesken.** Yhteisistä ratkaisuista, joita on analysoitu useasta eri näkökulmasta, tulee vähemmän erehtyviä, ja ne kattavat useamman käyttötapauksen kerrallaan. Tämä vähentää ratkaisujen refaktorointi. Joustavuus ja asenne muuttuviin vaatimuksiin paranee ja kokonaisprosessia on helpompi seurata ja ymmärtää, kun kaikilla on jatkuvasti yhtenäinen käsitys isosta kuvasta ja siitä, mihin suuntaan suunnittelu on kehittymässä. Koko tiimi kykenee seuraamaan esiin nousseiden suunnittelukysymysten etenemistä. Koko suunnitteluun liittyvä prosessi tulee olla läpinäkyvä koko tiimille, mikä parantaa lopputulosta ja auttaa iteraatiota pysymään raiteillaan [72, 87]. **Suunnittelijat viettävät lopulta enemmän aikaa suunnittelun parissa, mutta kehittäjien on osallistuttava suunnitteluratkaisujen kehittämisen lisäksi aktiivisesti suunnittelun katselmointiin, ja palautteen antamiseen.** Näin kehittäjien ajattelu laajenee laadun ja käyttäjäkokemuksen puolelle, ja suunnittelijalla on vähemmän huomautettavaa kehityksen tuotoksista.

Suunnittelu jaetaan paloihin ja tehdään mahdollisimman myöhään, jotta suunniteltaessa tiedetään kaikki viimeisimmät siihen vaikuttavat tekijät. [73, 76] Hyvät suunnittelijat mukauttavat kehitykseen toimitettavia suunnitelmiaan tarpeiden mukaan. [90] Tästä syystä suunnittelua ei kannata tehdä yksityiskohtaiseksi turhan aikaisin. Ketterissä me-

netelmissä ei välttämättä ole yhtään vähempää suunnittelutyötä kuin muilla menetelmillä läpiviedyssä projektissa [76], mutta ketterissä menetelmissä pyritään minimoimaan suunnittelutyön kertautuminen. Projektin alussa kuitenkin suunnittelutyötä voi olla toisinaan hyvä tehdä yksityiskohtaisemmin, jotta saadaan luotua yleinen malli yksityiskohdista [45], mutta mitä pidemmälle projektissa edetään, sitä vähemmän toiminnallisuuksien pieniin yksityiskohtiin tulisi kiinnittää huomiota suunnittelussa. Projektin aikana on hyvä esittää kysymys ”Mikä on vähintä, mitä tarvitsee tietää ennen toiminnallisuuksien rakentamista?” [45] On tärkeämpää saada jotain aikaiseksi kuin suunnitella [76], ja tunnistaa, mitkä tehtävät tuovat arvoa lopputuotteelle ja lopulta loppukäyttäjille. Tätä voi avata itselle kysymällä ”Mikä on yksinkertaisin mahdollinen tapa kohdata käyttäjän tarpeet [45] ja tuottaa käyttäjälle arvoa?” **Suhteellisesti vähemmän arvoa tuottavat tehtävät jätetään tekemättä, ja panostetaan enemmän arvoa tuottaviin toiminnallisuuksiin.** Kun yksinkertainen ja arvokas tuote on paikoillaan, voidaan kehitystä helpommin jatkaa eteenpäin [45].

Suunnittelua ei ole välttämätöntä aloittaa navigaation tai etusivun suunnittelusta, vaikka se suunnittelijoille onkin luonnollista. Monesti voi olla järkevämpää tehdä nämä vasta, kun muita ominaisuuksia on saatu rakennettua ja ymmärrys kokonaisuudesta paranee. Suunnittelu tuleekin keskittää siihen, mikä on priorisoituna tuotteen kehitysjonolle. [45] **Kun suunnittelu on valmis, suunnittelija ei irrota käsiään työstään**, vaan työstäminen jatkuu kehitykselle annettavana tukena kehitysiteraation aikana. [81]

Ketterän työskentelytavan onnistuminen riippuu siitä, miten tiimi on onnistunut muokkaamaan menetelmää itselleen sopivaksi ja tehokkaaksi. Tämä vaatii tiimiltä kokemuksesta ja keskinäistä luottamuksesta, sekä tiimin jäsenten kykyä työskennellä lähekkäin [72]. **Projektin alussa voidaan määritellä tehtävät, joita tiimi projektin aikana tekee, sekä vaiheittaiset tavoitteet, joita kohden tiimi pyrkii.** Tavoitteita asetetaan myös suunnittelijoiden aktiviteeteille [87], kuten kuinka monelle käyttäjälle käyttäjätutkimus tehdään, kuinka usein tuotetta testataan, mikä on MVP, mitkä tehtävät kuuluvat tiimille ja mitkä kuuluvat tiimin yrityksen muille henkilöille. Esimerkiksi palvelumuotoilu voi olla erillinen, tiimin ja projektin ulkopuolinen tehtävät. Kun suunnitteluja kehitystyön ulkopuoliset tehtävät eivät rasita tiimiä, jää tiimille aikaa tuottaa itse projektille arvoa. Toimittava yritys voi kuitenkin yhdessä pohtia tehtävien kohdentamista projektille ja sen ulkopuolisiin asiantuntija tehtäviin asiakkaan kanssa, ja varmistella paikkaansa tulevissa kehitysprojekteissa.

7.5 Muita hyviä käytäntöjä

Suunnittelijoihin, tiimiin ja prosessiin liittyevien käytäntöjen ja toimintatapojen lisäksi on muitakin verkkolähteissä usein esiin tulleita käytäntöjä, jotka vaikuttavat ketteryyden onnistumiseen eri abstraktiotasoilla. Tällaisia ovat esimerkiksi iteraation mittaaminen,

dokumentointi, testaaminen, suhtautuminen iteraatiossa epäonnistumiseen ja toimittavan yrityksen toimintaan. Näitä aiheita käsitellään seuraavissa kappaleissa.

7.5.1 Iteraation mittaaminen

Iteraation onnistumista ja projektissa etenemistä voidaan mitata. Scrumissa sprintin aikana mitataan työmäärää, jonka tiimi tekee sen aikana (*engl. velocity*). Tämän mittaaminen ei kuitenkaan auta tiimiä keskittymään siihen, mitä ollaan oikeasti saatu aikaiseksi, vaan se tähtää saamaan ulos mahdollisimman paljon uusia toiminnallisuuksia iteraation aikana. Tällöin tiimiltä saattaa kadota ymmärrys, mitä ”valmis” tarkoittaa. **Määrää tärkeämpää on julkaista toiminnallisuuksia**, joita käyttäjät käyttävät, joilla on arvoa, ja jotka parantavat asiakkaan liiketoimintaa. [82] **Ketterissä projekteissa edistymistä pitäisi mitata ensisijaisesti toimivan ohjelmiston kautta.** [76, 78]

7.5.2 Dokumentointi

Monesti on ymmärretty, että ketterissä menetelmissä ei dokumentoida mitään. Agile Manifesto [64] kuitenkin sanoo, että ”Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota”, mikä ei tarkoita, että dokumentaatio sivuutetaan. Dokumentaatiota voidaan tehdä, mutta vain minimaalinen määrä. ”Suunnittelijoiden tulisi keskittyä suunnitteluun ja kehittäjien kehitykseen.” [45, 67] Miksi palkata suunnittelijat kirjoittamaan dokumentaatiota, kun he ovat parempia suunnittelussa? **Dokumentaation kirjoittaminen muodostuu helposti pullonkaulaksi etenkin suunnittelutyölle** [45, 67].

Toisinaan asiakas vaatii dokumentaatiota, vaikka se haluaa olla ketterä. Ongelma muodostuu, jos asiakasyritys arvottaa dokumentaatiota enemmän kuin panostusta suunnitteluun ja kehitykseen. Tässä on kuitenkin lopulta kysymys asiakkaan arvoista ja valinnoista [45, 67], joihin voi pyrkiä vaikuttamaan kouluttamalla asiakasta. Asiakkaalta voidaan kysyä miksi ja mihin hän tarvitsee dokumentaatiota, millä tavalla dokumentaatio tuottaa tälle lisäarvoa ja kuinka usein ja kuka dokumentaatiota myöhemmin lukee? Kuka ja miten dokumentaatiota ylläpidetään? Asiakkaan saattaa olla vaikea vastata myös kysymykseen, mitä siitä seuraisi, jos dokumentaatiota ei tehtäisikään, jos asiakkaalla ei ole selkää tarvetta dokumentaatiolla. **Jos kattava dokumentaatio päätetään lopulta kirjoittaa, se tulisi tehdä vasta projektin lopuksi**, jotta suunnittelu- ja kehitysiteraatiot eivät hidastuisi, eikä dokumentaation kirjoitustyö kertaudu muutosten myötä. Dokumentaation tarpeellisuudesta ja kattavuudesta on hyvä keskustella asiakkaan kanssa jo sopimusvaiheessa.

Verkkokeskusteluissa [45, 67] ja haastatteluissa todettiin, että käyttäjätarinat ovat tärkeimpiä asioita dokumentoida, jotta kuka tahansa voisi käydä läpi tuotteen kehitysjonon ja ymmärtää kokonaisuuden myös jälkikäteen. Käyttötapaukset ja käyttäjätarinat ovat

nopea ja tehokas tapa kirjata tuotteen vaatimukset kaikkien näkyville. [90] Artikkelissaan Six [91] muistuttaa, että projektissa on tärkeää dokumentoida myös poikkeustapaukset, jotta ne tulevat suunnitteluratkaisuissa katettua. Yleensä muistetaan suunnitella ns. onnellinen reitti (*engl. happy path*) hyvin, mutta vaihtoehtoiset toimintatavat ja poikkeukset jäävät kunnolla miettimättä, ja heikentävät käyttäjäkokemusta. Eräässä Internet-keskustelussa [66] huomautetaan, että tärkeimmät tekniset päätökset tulisi dokumentoida. Haastatteluissa todettiin, että palaverimuistiinpanot ovat tärkeä dokumentaatio tehtyjen päätösten osalta. **Käyttötapausten, käyttäjätarinoiden, persoonien ja palaverimuistiinpanojen lisäksi tulisi dokumentoida tietokantakaavio, korkean tason arkkitehtuurimalli, tiedonkulkukaavio tärkeimmistä tietovirroista ja käyttöönottodokumentti mielellään kuvamuodossa.** [haastattelut, 66]

7.5.3 Testaaminen

Haastatteluissa kaikki yritykset sanoivat jättävänsä käyttäjätestauksen pääasiassa väliin, koska haastateltavien mielestä testaaminen sopii huonosti iteratiiviseen prosessiin. Jonkinlaista käyttäjätestausta tehdään alussa prototyypillä, mutta testaaminen ei ole säännöllistä eikä jatkuvaa. Haastatteluissa käyttäjätestaaminen nähtiin suunnittelijoiden työksi, kun taas kehittäjät testaavat ainoastaan koodia ja järjestelmien integraatiota.

Käyttäjätestaus on kuitenkin iteratiivinen prosessi [96, 90]**, ja testaamista tulisi tehdä säännöllisesti.** Etenkin projektin alussa käyttäjätestaus on tärkeää, koska muutosten kustannukset ovat vielä pieniä. [96] Testit voidaan suunnitella jo ennen kuin oikeastaan tiedetään, voidaanko asiaa testata [83] perustuen siihen, mikä on tuotteen visio, ja mitkä ovat tuotteelle asetetut käyttäjäkokemustavoitteet. Konseptivaiheessa on tärkeintä testata hypoteeseja ja oletuksia, joista koetaan eniten epävarmuutta [93].

Käyttäjätestausta voidaan tehdä viikon tai kahden välein iteraation päättyessä pienissä sessioissa. [13, 83, 84, 96] Käytettävyystestaus voidaan ottaa osaksi tuotoksen hyväksymistestausta [13], joka on yksi suunnittelijoiden merkittävistä tehtävistä. Testattavien käyttäjien määrä ei tarvitse olla suuri, sillä jo neljällä käyttäjällä löydetään vakavimmat virheet kustannustehokkaasti [96]. Nielsen [32] on argumentoinut viiden testikäyttäjän puolesta. On huomioitavaa, että mitä useammin käyttäjätestausta tehdään, sitä pienemmällä osanottajamäärällä testausta tarvitsee tehdä. **Huonoin vaihtoehto olisi jättää kokonaan testaamatta**, jolloin ei voida varmistua, että projekti tuottaa loppukäyttäjälle varmasti arvoa. Käyttäjätestauksen tekeminen vasta tuotteen valmistuttua on myös huono vaihtoehto, koska siinä vaiheessa muutoksia ei voida enää sisällyttää projektiin, eikä saadulla palautteella anneta tuotteelle enää lisäarvoa.

Koko tiimin tulisi osallistua käyttäjätestaukseen, kuten käyttäjätutkimukseen ja suunnitteluun, jotta kaikki ymmärtävät käyttäjien ongelmat ja palautteen [81, 87, 90,

96], ja korjauksia tuotteelle voidaan tehdä perustuen käyttäjien antamaan palautteeseen, johon myös kehittäjät voivat antaa ratkaisuvaihtoehtoja. Tämä jälleen korostaa tiimin yhteisvastuusta projektista ja sen lopputuotteesta.

7.5.4 Ketteryydessä epäonnistuminen ja jatkuva parantaminen

Yksi vaikeimmin omaksuttavia periaatteita ketteryydessä on epäonnistumisen hyväksyminen, vaikka ketteryyden yksi perussäännöistä on ”*Epäonnistu nopeasti ja epäonnistu usein*”. Epäonnistuminen on vaikea aihe sekä tiimin, että organisaation tiimin ympärillä, mutta myös asiakkaalle. Etenkin ketterien menetelmien ollessa asiakkaalle vieraita, pelko voi ohjata asiakkaan toimintaa. Tästä voi seurata muun muassa se, että tiimin ei anneta todellisuudessa olla itseohjautuvia, vaan johto tai asiakas sekaantuvat projektiin ujuttamalla omia prosessimalleja mukaan, tai vaatimalla kurinalaista Scrumin seuraamista [52]. Ketteryyden arvojen väärinymmärtäminen voi johtaa projektin ja ketteryyden epäonnistumiseen, jolloin ketteryyden ei ajatella istuvan kyseiseen projektiin tai kyseiselle organisaatiolle. Tiimiltä vaaditaankin kypsyyttä hallita sovellettua menetelmää itseohjautuvasti [52].

Tiimi voi epäonnistua arvon tuottamisessa asiakkaalle, tai se voi korostaa liikaa työkaluja tai prosesseja oikean kommunikaation sijaan. Tiimi saattaa keskittyä liikaa siihen, mitä he itse haluavat tehdä, ja jättää asiakkaan tarpeet toissijaisiksi, mutta nämä eivät ole ketterän menetelmän tai filosofian syitä. **Ketterillä prosesseilla ei voida taata tiimin ja projektin onnistumista**, sillä kyse on ennen kaikkea ihmisistä, ja heidän kyvystään toteuttaa ketteryyden periaatteita ja sovittuja toimintatapoja. **Ketteryys voi kuitenkin auttaa tiimiä epäonnistumaan ajoissa**, ja siten minimoimaan kustannuksia ja hävikkiä [52].

Ketteryydessä, suunnittelussa ja toteutuksessa on siis lupa epäonnistua, vaikka se onkin pelottava ajatus [76]. **Epäonnistumisen riskiä voidaan minimoida aikaisella käyttäjätestauksella ja testaamalla projektin aikana pieniäkin muutoksia** esimerkiksi A/B-testauksella. Ideana on tehdä vähäriskisiä kokeiluja, ja mitata niiden toimivuutta heti [95]. Nopeiden ratkaisuideoiden kokeilu johtaa toisinaan epäonnistumisiin ja vääriin lopputuloksiin. Nopea virheen tai epäonnistumisen toteaminen testauksella on ratkaisevan tärkeää, jotta virhe saadaan korjattua nopeasti, eikä se aiheuta kerrannaisvaikutuksia tai vahingoita tuotetta.

Ketteryyden periaatteena on jatkuva parantaminen, josta syntyy muutostarpeita olemassa olevaan tuotteeseen sekä tiimin toimintatapoihin. Kaikki muutostarpeet eivät ole epäonnistumisia, vaan **projektin edetessä ymmärrys vaatimuksista, ratkaisuvaihtoehdoista ja ratkaisujen vaikutuksista lisääntyvät.** Siksi on sekä asiakkaalle, organisaatiolle että tiimille tärkeää olla avoimia muutoksille. Niihin voidaan varautua jo

projektin alussa suunnittelemalla olemassa oleva arkkitehtuuri ja rakenne sellaiseksi, että niihin on helppo tehdä muutoksia ja lisätä toiminnallisuuksia. Epäonnistumisten takia ei pidä pysähtyä, vaan pitää ottaa opiksi, yrittää uudelleen, ja parantaa ongelmakohtia seuraavalla iteraatiolla tai jatkoprojektissa. Ensimmäinen yrittämä ei ole täydellinen kenelläkään, ja oikean toimintatavan löytäminen vaatii harjoittelua.

"*In my opinion, there is no failure, it is only feedback.*
It indicates that current approach is not effective.
People start transformation or adoption with some expectations.
When they don't see results they call it a failure.
It is a way to communicate that it did not meet the expectation or objectives.
What I encourage is focus on the learning and move on, do something different. "
– Epin John Poovathany [52]

Jatkuva parantaminen on mahdollista vain kommunikoinnin ja palautteen avulla. Hyvä kohta katselmoida tiimin ja projektin onnistumista on retrospektiivi iteraatioiden välissä, vaikkei menetelmäksi olisikaan valittu Scrumia [52]. Retrospektiiviä ei tarvitse pitää jokaisen iteraation jälkeen, mutta tiimin kehityksen ja oppimisen kannalta niitä olisi hyvä pitää säännöllisin väliajoin

7.5.5 Ohjelmistoyrityksen oma toiminta

Organisaation rooli projektin onnistumisen kannalta on merkittävä. Kulttuuri ja asenne organisaatiossa ovat ratkaisevia tiimin keskinäisen ja asiakkaan kanssa tehtävän yhteistyön onnistumisen kannalta. Positiivinen suhtautuminen suunnitteluun hyödyntää kaikkia osapuolia. **Käyttäjäkeskeisen suunnittelun ja ketteryyden pitää olla osa koko organisaation kulttuuria** [87, 25]**.** Organisaatioissa, joissa on ohjelmistokehityskeskeinen kulttuuri, toteutetaan ominaisuuksia, joiden tarpeita eikä käyttäjäkokemusta ymmärretä. [87] **Ketterästi toimivan organisaation tulee toiminnoillaan tukea suunnittelutyön integroitumista ketteryyteen ja mahdollistaa itseohjautuva ketterä toiminta.**

Ketterä organisaatio on avoin, läpinäkyvä, inhimillinen, motivoiva, itseohjautuvuuteen kannustava ja jatkuvasti mukautuva [78]. Organisaatio voi näitä ominaisuuksia laiminlyömällä vaikeuttaa todellisen ketteryyden käyttöönottoa koko organisaation tasolla, joka taas periytyy projekteihin. **Usein ketteryydessä epäonnistuminen nähdään johtuvan ylimmän johdon vakavasta sitoutumisen puutteesta, ja keskijohdon muutosvastarinnasta** [52]**.** Jos osa organisaatiosta työskentelee ketterästi, ja osa tarkkaan suunnitelmaan perustuen, tulee näiden yhteistyöstä ongelmallista. [25] Muutos ketteryyttä kohden saattaa epäonnistua ja ylimmän johdon sitoutuminen entisestään heiketä.

Ketteryyden omaksuminen organisaatioon vaatii ennen kaikkea luottamusta, läpinäkyvyyttä, sitoutumista ja muutosta koko organisaatiokulttuurissa. [52]

"*It's not Agile that is failing the business,*
just the people in the business failing to be Agile."
– Guy Maslen [52]

"*People say 'Agile failed' because people like to blame things instead of themselves.* "
– Michael Fritzius [52]

Haastatteluissakin tuli ilmi, että suunnittelijat kokevat ajan olevan riittämätön, mikä suunnittelutyölle resursoidaan. Van Weerdenburg [92] sanookin, että **projektit tulee aikatauluttaa ja resursoida realistisesti, ei optimistisesti.** Suunnittelijoita ja kehittäjiä ei tule painostaa tekemään työmääräarvioita eksaktisti, koska asiantuntijat eivät kykene arvioimaan työmäärää kaikkien tulevien muutosten osalta. **Tiimiä ei tule pakottaa venymään yli suorituskyvyn**, koska liika työmäärä tappaa luovuuden, kyvyn mukautua ja hahmottaa kokonaisuuksia. Moni näistä paineista ja vaatimuksista tulee kehitystiimin ulkopuolisilta sidosryhmiltä, ja kehitystiimin oman organisaation on hyvä varautua näihin paineisiin ja vaatimuksiin jättämällä aikaresursointiin hieman ilmaa, jotta muutokset eivät ylikuormita tiimiä.

8. KOONTI HYVISTÄ KÄYTÄNNÖISTÄ JA TOIMINTATAVOISTA

Tässä luvussa on koottu haastattelututkimuksen ja verkkoaineistoanalyysin tulokset yhteen perustuen diplomityölle asetettuihin tutkimuskysymyksiin.

8.1 Käyttäjäkeskeinen suunnittelu osana ketterää ohjelmistokehitystä

Diplomityölle asetettu ensimmäinen tutkimuskysymys oli, miten käyttäjäkeskeinen suunnittelu otetaan huomioon osana ketterää ohjelmistokehitysprosessia. Kysymyksellä haluttiin selvittää erityisesti, miten kehitystiimistä saadaan muodostettua yhtenäinen, eikä käyttäjäkeskeiseen suunnitteluun erikoistuneet asiantuntijat olisi kehitystiimin ulkopuolinen osa. Tarkoituksena oli myös selvittää, millä keinoilla käyttäjäkeskeiseen suunnitteluun liittyvät tehtävät saadaan koettua arvokkaiksi.

Käyttäjäkeskeisen suunnittelun arvottaminen lähtee yritysorganisaatiosta, jossa käyttäjäkeskeinen suunnittelu tulee olla osa koko organisaation kulttuuria ja tunnustettu ja arvostettu osa-alue yrityksen strategiaa. Kuten kuvassa 10 esitetään, että ketteryyden tulisi lähteä organisaation ylimmältä tasolta kohti alempia tasoja, tulisi käyttäjäkeskeiseen suunnitteluun suhtautua organisaatiossa samalla tavalla. Jos organisaation ylin johto ei tue toimillaan käyttäjäkeskeistä suunnittelua ja tunnusta sen arvoa liiketoiminnalle, ei sille ole valmiuksia toteutua alemmillakaan tasoilla.

Projektissa käyttäjäkeskeisen suunnittelun mukaanotto lähtee kehitystiimin resursoinnista, joka tulisi tehdä kompetenssien perusteella, eikä rooleittain. Tämän tarkoituksena on jakaa eri projektitehtäviä kehitystiimin kiinnostusten mukaan, eikä osoittaa tiettyjä tehtäviä tietyille rooleille. Kehitystiimi itseohjautuvasti ja omien sisäisten resurssien mukaan jakaa tehtäviä kaikkien tekijöiden kesken esiselvityksestä lähtien. Tällä pyritään siihen, että kompetensseista riippumatta kaikki kehitystiimin jäsenet osallistuisivat projektin eri vaiheiden tehtäviin, eivätä jäsenet, joilla ydinkompetenssi on suunnittelussa, joutuisi ainoina tekemään suunnittelutehtäviä. Lisäksi pyritään siihen, että kaikille muodostuu selkeä kuva asiakkaan ja käyttäjien tarpeista, visiosta ja isosta kuvasta, sekä kaikki ymmärtävät, mihin suunnitteluratkaisut perustuvat. Kun koko kehitystiimi osallistuu erityyppisiin projektin tehtäviin, kehitystiimistä muodostuu yhtenäinen ja vastuu kokonaisuudesta jaetaan koko kehitystiimin kesken. Tällöin käyttäjäkeskeinen suunnittelu, ja sitä myöden tuotteen laatu ja käyttäjäkokemus, ovat koko kehitystiimin vastuul-

la, eikä ainoastaan suunnittelijoiden. Myös asiakkaan on helpompi hyväksyä, että suunnittelutyö on osa koko projektin elinkaarta, eikä suunnittelutyötä voida leikata projektista pois yhtä helposti kuin kehitystiimeissä, jotka koostuvat rooleista.

Riippumatta siitä koostuuko tiimi rooleista vai kompetensseista, suunnittelijoilla ei tulisi olla liian montaa projektia työn alla saman aikaisesti. Jos suunnittelija ei ole muun kehitystiimin tavoitettavissa, vähentää se suunnittelijan ja käyttäjäkeskeisen suunnittelun integroitumista kehitystiimiin ja projektiin. Kun suunnittelijalla on monta projektia työn alla samanaikaisesti, riski pullonkaulojen muodostumiseen kasvaa. Pullonkaulat eivät vaikuta ainoastaan suunnittelijoiden työhön, vaan aiheuttavat viivästyksiä kehitystiimin muihin tehtäviin. Viivästyksiä muodostuu mahdollisesti tällöin suunnittelijan muissakin projekteissa. Viivästyksistä johtuen käyttäjäkeskeinen näkökulma saatetaan projektissa ohittaa, jolloin suunnittelutyön integroituminen heikkenee. Kun suunnittelijoiden ja muun kehitystiimin esteetön kommunikaatio mahdollistetaan, integroituminen helpottuu, ongelmat on nopeammin ja helpommin ratkaistavissa ja lopputuotteen laatu paranee.

Tuotteen kehitysjonolle tulisi laittaa myös käyttäjäkeskeiseen suunnitteluun liittyvät tehtävät näkyväksi koko kehitystiimille. Koko kehitystiimille tulisi olla läpinäkyvää, mitä projektissa ollaan tällä hetkellä tekemässä, mitä on tekemättä ja mitä tehdään seuraavassa iteraatiossa. Kehitysjonolla olevat käyttäjäkeskeiseen suunnitteluun liittyvät tehtävät ovat myös asiakkaille helpompi ymmärtää, koska ne kuvaavat isompia kokonaisuuksia, ja näin keskustelu kehitystiimin ja asiakkaan välillä helpottuu.

8.2 Käyttäjäkeskeisen suunnittelun integroitumista tukevat käytännöt

Toinen tutkimuskysymys oli, mitkä käytännöt tukevat suunnittelu- ja kehitystyön integraatiota ketterässä ohjelmistokehitysprosessissa. Tämän kysymyksen tarkoituksena oli selvittää yksittäisiä käytäntöjä ja toimintatapoja, jotka mahdollistavat käyttäjäkeskeisen suunnittelun sujuvan integraation ketterään prosessiin. Tähän kappaleeseen on kerätty tärkeimpiä käytäntöjä integroitumisen onnistumisen kannalta.

Kehitystiimi muodostetaan kompetensseista

Käyttäjäkeskeinen suunnittelu tulisi olla luonnollinen osa ketterää ohjelmistokehitysprosessia. Keinoja integroida käyttäjäkeskeisyys ketterään prosessiin on muun muassa edellä mainittu kompetenssien resursointi kehitystiimiin roolien sijasta. Tällä tavalla painotetaan kehitystiimin asiantuntijoiden osaamisalueita, ja annetaan mahdollisuus toimia tehtävissä, jotka eivät kuulu asiantuntijan ydinkompetenssiin, mutta joista tämä on kiinnostunut. Rooliajattelu ajaa asiantuntijat lokeroihin ja ottamaan vastuun ainoas-

taan omista tehtävistään. Kompetenssiajattelu mahdollistaa kokonaisvastuun jakautumisen koko kehitystiimille yhdessä.

Kehitystiimi toimii itseohjautuvasti yhdessä

Tehtävien jako, työkalujen ja menetelmien valinta ja näihin liittyvä päätöksenteko perustuvat joustavuuteen ja kehitystiimin itseohjautuvuuteen. Itseohjautuva kehitystiimi sitoutuu projektiin liittyviin päätöksiin, koska ovat itse saaneet vaikuttaa niihin. Kuten edellisessä kappaleessa kuvataan, koko kehitystiimi tulisi osallistua kaikkiin projektiin liittyviin tehtäviin, vaikka ne perinteisesti kuuluisivatkin suunnittelijoille. Itseohjautuvassa kehitystiimissä tehtävät ohjautuvat luonnollisesti tekijöilleen. Itseohjautuvuus ja yhteisvastuu onnistuvat vain, jos suunnittelijat ovat täysin integroituneita osaksi muuta kehitystiimiä.

Suunnittelijalla tulisi olla pari, jolla on samaa kompetenssitaustaa. Suunnitteluasiantuntijat pystyvät tukemaan toisiaan muun muassa sparraamalla toisiaan ja antamalla ratkaisuista toisilleen nopeaa palautetta. Kahden suunnittelijan avulla saadaan aikaiseksi laadukkaita ja innovatiivisia ratkaisuja helpommin kuin yhdellä suunnittelijalla. Toiseksi suunnittelijaksi voidaan laskea myös front end -kehittäjä, jolla on osaamista käyttöliittymistä.

Panostetaan kehitystiimin kommunikaatioon

Yhteissuunnittelu ja kommunikointi tiimin kesken ovat tiukasti sidoksissa toisiinsa. Kehitystiimin sisäiseen kommunikaatioon tulisi panostaa jo ennen projektin alkamista. Kehitystiimille voidaan järjestää yhteistä tekemistä, kuten illallinen, jotta asiantuntijat tutustuvat toisiinsa. Kun asiantuntijat tuntevat toisensa, madaltuu kommunikaatiokynnys ja kommunikaatiosta tulee välitöntä ja avointa. Lisäksi kehitystiimille tulisi mahdollistaa jatkuva kasvokkain käytävä kommunikaatio, mieluiten yhteisen työskentelytilan myötä.

Allokoidaan suunnittelutehtäville tarpeeksi aikaa

Suunnittelukompetenssille tulee allokoida tarpeeksi aikaa. Organisaation pitää hyväksyä, että suunnittelutyön määrässä on projektin aikana vaihtelua. Hetkittäisen vähemmän työmäärän takia suunnittelukompetenssia ei kannata leikata projektilta pois tai kokonaisallokaatiota pienentää. Projektin alussa suunnittelutyö on kaikkein eniten aikaa vievää, sillä tuolloin suunnittelijoiden tulee tehdä tarpeeksi esitutkimusta, jotta projektille voidaan luoda visio. Vision luomisen lisäksi suunnittelijat määrittelevät projektin ison kuvan, luovat konseptin ja mahdollisesti testaavat sen ennen varsinaisen toteutuksen aloittamista. Tähän kaikkeen ei välttämättä riitä yhden iteraation pituinen jakso, eli viikko tai kaksi, vaan koko esiselvitys voi viedä kolme tai neljäkin viikkoa projektin koosta riippuen.

Kuten aiemmin huomautettiin, työmääräresursoinnissa tulisi huomioida vaihtelevuus. Huono työmääräresursointi saattaa johtaa kiireeseen, joka voi johtaa pullonkaulojen syntymiseen. Suunnittelijoiden kiirettä lisää usean projektin samanaikainen työstäminen. Tämä heikentää myös käyttäjäkeskeisen suunnittelun integroitumista ohjelmistokehitykseen. Resursoimalla tarpeeksi esiselvitykseen ja ison kuvan varmistumiseen, voidaan vähentää pullonkaulojen riskiä. Kiire tilanteessa tulisi keskittyä arvoa tuottaviin toiminnallisuuksiin, ja jättää asiakkaalle vähemmän arvoa tuottavat tehtävät tekemättä. Pullonkauloja voidaan ehkäistä myös määrittelemällä projektin aluksi suunnitteluperiaatteet selkeästi, jolloin kehityksen voidaan ohjata eteenpäin pelkästään keskustelemalla.

Kehitys- ja suunnitteluprosessia ei vakioida

Ketterää ohjelmistoprosessia ja siihen liittyvää suunnitteluprosessia ei ole tarpeen vakioida. Kuten haastatteluissa todettiin, jokaisella projektilla on erilaiset tarpeet ja tavoitteet, jotka vaikuttavat valittuihin menetelmiin, työkaluihin ja iteraatioiden pituuteen. Vakioiminen ei myöskään nojaa ketteryyden periaatteisiin, joiden mukaan menetelmien ja tehtävien osalta tulisi olla joustava, ja jokaisen kehitystiimin tulisi toimia itseohjautuvasti. Yritys tai organisaatio voi kuitenkin määritellä viitekehityksen jolla määritellään toimintaperiaatteita ja ohjeistuksia ketterään työskentelyyn.

Suunnittelu tehdään juuri ennen kehitystä

Suunnittelua pyritään tekemään yhtä tai kahta iteraatiota kehitystä edellä, tai muuten juuri ennen kehitystä. Tämän tarkoitus antaa suunnittelijoille viimeisin toiminnallisuuteen liittyvä tieto suunnitteluratkaisujen tueksi, ja näin minimoida toiminnallisuuden uudelleensuunnittelu. Edellinen ei tarkoita, että suunnittelu tehtäisiin kokonaan kahden edeltävän iteraation aikana, vaan tuolloin toiminnallisuus viimeistellään. Toiminnallisuuksia voidaan hahmotella ja luonnostella pitkin projektia korkealla tasolla. Tämä on osa isoon kuvaan liittyvää hallintaa ja työtä.

Suunnittelukompetenssi pidetään mukana koko projektin ajan

Kuten edellisessä kappaleessa tuotiin esille, suunnittelija tai suunnittelijakompetenssin omaava asiantuntija tulisi pitää mukana projektissa koko sen läpiviennin ajan. Käyttäjäkeskeiseen suunnitteluun liittyvät työt voivat vähentyä projektin loppua kohden, jolloin suunnittelijoille allokoitua aikaa voidaan vähentää, mutta sitä ei tule leikata kokonaan pois.

Testaamista tehdään säännöllisesti

Testaaminen voidaan tehdä pienissä kokonaisuuksissa muutaman iteraation välein. Käyttäjätestauksen en tarvitse olla iso tilaisuus, vaan suurimmat ongelmat löydetään jo

neljällä tai viidellä käyttäjällä. Testaamisesta kerrotaan seuraavassa kappaleessa enemmän.

Testauksesta seuraa väistämättä muutoksia, ja ne ovat luonnollinen osa iteratiivista prosessia. Tästä syystä iteraation sisältöä ei lyödä tiukasti lukkoon, eikä siihen mahduteta liikaa tehtäviä. Iteraatioon tulee jättää tilaa hioa ratkaisuja, testata tai kerätä muuten palautetta, ja tehdä korjauksia.

Epäonnistuminen johtaa toiminnan parantamiseen

Epäonnistumista ei pidä pelätä, vaan siitä pitää oppia. Ketteryyden periaatteena onkin jatkuva parantaminen. On todennäköistä, ettei kehitystiimi pysty parhaimpaansa heti projektin alussa, vaan yhdessä toimiminen ja oikeiden toimintatapojen löytäminen vaativat harjoittelua. Kehitystiimin tulee toimia yhdessä ongelmien esiin nostamiseksi ja niiden korjaamiseksi. Myös hyvin toimivan kehitystiimin tulisi ajoittain pitää retrospektiivi ja tarkastella omaa toimintaansa, ja etsiä parempia käytäntöjä ja toimintatapoja.

Projektin tehtävistä saadut tuotokset riittävät dokumentiksi

Dokumentointi ei ole ketteryyden vihollinen, eikä dokumentointia tulisi jättää kokonaan tekemättä. Projektin eri vaiheiden ja menetelmien tuotoksia voidaan käyttää dokumentaationa erikseen kirjoitetun dokumentaation sijaan. Dokumenteiksi käyvät projektin alussa luodut käyttäjätarinat, käyttötapaukset ja skenaariot, iteraatioiden aikana tehdyt tussitaulupiirrokset, jotka on valokuvattu, rautalankamallien lisäksi, sekä projektin aikana kirjoitetut palaverimuistiinpanot.

Asiakas koulutetaan kehitystiimin tapaan olla ketterä

Jos asiakas ei ole tuttu ketterien toimintatapojen kanssa, kannattaa projektin aluksi asiakasta kouluttaa ketteryyteen ja kertoa, mitä se tarkoittaa ja miten asiat etenevät, sekä mitä asiakkaalta projektin aikana odotetaan. Nämä asiat on hyvä käydä läpi, vaikka asiakas tuntisikin ketterän lähestymistavan, sillä jokaisella yrityksellä ja projektilla on oma tapansa toimia ketterästi. Asiakkaaseen liittyvät odotukset, kuten sitoutuminen päätöksentekoon ja palautteen antaminen, on tärkeimpiä asiakkaan kanssa sovittavia asioita projektin aluksi, koska päätöksenteon hetkellä poissa oleva asiakas hidastaa projektia, ja ilman palautetta tuotetta ei voida kehittää asiakkaan toivomaan suuntaan, jolloin tuotteen laatu kärsii.

Asiakas vastaa kehitysjonon priorisoinnista

Asiakas on taho, joka ostaa tuotteen, jolloin asiakkaan tulisi tehdä siihen liittyvät priorisointipäätökset. Tästä syystä kehitystiimin tulisi antaa vastuu kehitysjonon priorisoinnista asiakkaalle. Kehitystiimin tulee auttaa priorisoinnissa asiakasta ja osallistua siihen

liittyvään keskusteluun, mutta asiakkaan tekee lopullisen päätöksen. Projektin seurantaan ja toteutukseen liittyvä taulu, kuten Kanban-taulu, tehdään kuitenkin kehitystiimin tarpeita silmällä pitäen, ja tiimi kuljettaa siihen liittyviä tehtäviä vaiheelta toiselle.

8.3 Arvoa tuottavat suunnittelutehtävät

Viimeinen diplomityölle asetettu tutkimuskysymys oli, mitkä suunnittelutehtävät koetaan tuottavan eniten hyötyä prosessin ja lopputuloksen kannalta. Tähän tutkimuskysymykseen ei voi vastata listalla suunnittelutehtäviä, sillä kuten haastatteluissa ilmeni jokainen projekti ja asiakas on erilainen, jolloin myös projektille ja tuotteelle asetetut tavoitteet vaihtelevat suuresti toisistaan.

Projektiin mukaan otettujen suunnittelutehtävien tulee tuottaa projektille ja lopputuotteelle lisäarvoa. Suunnittelutehtävät valitaan projektin painopisteen ja fokuksen mukaan. Toisinaan projekti koskee tuotetta, joka on jo tuotannossa, eikä tätä varten välttämättä tarvita laajaa käyttäjätutkimusta, jolloin esitutkimusvaihe jää pienemmäksi kuin projektissa, jossa tehdään täysin uusi tuote. Esitutkimusvaiheessa on tärkeää ymmärtää asiakkaan liiketoiminta, projektin tavoitteet ja käyttäjien tarpeet. Näiden tutkimisiin koettiin parhaimmiksi menetelmiksi haastattelut ja työpajat, joissa keskustelun ohella on mahdollista osallistaa asiakasta tai loppukäyttäjää tuotteen suunnitteluun. Esitutkimusvaiheessa on koko projektin osalta tärkeä saada etenkin asiakas sitoutumaan projektiin, joka onnistuu parhaiten antamalla asiakkaalle mahdollisuus vaikuttaa suunnitteluun, ja tuntemaan, että asiakkaan mielipiteitä ja ajatuksia kuunnellaan. Työpajat ja haastattelut ovat erinomaisia menetelmiä tähän.

Kun projekti pääsee iteratiiviseen prosessiin, suunnittelutehtävät riippuvat pitkälti siitä, miten hyvin kehitystiimi kommunikoi keskenään. Pienin mahdollinen määrä ominaisuuden suunnittelua ennen sen toteutusta oli haastatteluiden perusteella ominaisuuden kuvaaminen kehittäjälle, tai sen tueksi piirretty nopea tussitaulupiirros. Suunnittelijoiden mukaan kaikista näkymistä ei ole tarvetta piirtää rautalankakuvia, koska kommunikaation sujuessa hyvin rautalankakuvat eivät tuo tuotteelle eikä projektille lisäarvoa. Päänäkymät tulisi kuitenkin piirtää, jotta asiakas voi ne hyväksyä ennen niiden toteutusta.

Vaikka haastattelujen perusteella testaaminen jätetään pitkälti väliin, on se kuitenkin arvokas menetelmä lopputuotteen kannalta. käyttäjätestaaminen on myös iteratiivinen prosessi, vaikka sitä ei sellaiseksi usein koeta. Testaaminen on erityisen tärkeää projektin alkuvaiheessa, jolloin muutokset vaativat vähiten uudelleensuunnittelua ja kehitystä, ja jolloin kustannukset ovat pienimmät muutoksille. Oikean konseptin varmistaminen käyttäjätestauksella projektin alussa on lopullisen tuotteen kannalta tärkein testausvaihe. Projektin aikana on hyvä testata pieniä ominaisuuksia ja toiminnallisia kokonaisuuksia

pitkin projektia, esimerkiksi kahden tai kolmen iteraation välein, jotta kehitystiimi saa jatkuvasti palautetta käyttäjiltä. Projektin lopuksi tehtävä testaus ei tuo tuotteelle lisäarvoa, sillä tällöin muutoksia ei tuotteelle enää tehdä.

9. JOHTOPÄÄTÖKSET

Työn tavoitteena oli löytää hyviä toimintatapoja ja käytäntöjä käyttäjäkeskeisen suunnittelutyön integroimiseksi ketterään ohjelmistokehitysprosessiin. Aihetta on käsitelty tutkimuksessa jo aiemmin, mutta hyviä käytäntöjä on koottu hajanaisesti muun tutkimuksen ohella. Kuusinen, K. kokosi vuonna 2015 väitöskirjaansa [25] viitekehyksen hyvistä käytännöistä, johon tässäkin diplomityössä on luvussa 4 viitattu. Aihe on saanut kiinnostuneen vastaanoton haastatelluissa yrityksissä. Yritykset ovat kiinnostuneita parantamaan omaa prosessiaan.

Ketteristä menetelmistä on tullut normi ohjelmistoprojektien läpiviemiseksi. Ohjelmistoyritykset tarjoavat, ja asiakkaat odottavat sitä. Käyttäjäkeskeinen suunnittelu ja käyttäjäkokemussuunnittelu ovat ohjelmistoyrityksille vakiintunut lähestymistapa tehdä projekteja, sillä sen arvo ymmärretään loppukäyttäjien, lopputuotteen että asiakkaan liiketoiminnan kannalta. Kuten Hyysalo [19] kirjassaan kirjoitti, hyvä käyttäjäkeskeinen suunnittelu on kannattavaa ja siihen panostaminen maksaa itsensä takaisin jo teknisen suunnittelun kustannuksissa, ja hyvin käyttäjille soveltuva tuote myy itsensä takaisin. Ohjelmistoyritykset haluavat hyötyä sekä ketteryydestä että käyttäjäkeskeisyydestä, mutta kokevat niiden integroimisessa paljon haasteita.

Tässä diplomityössä koottu ohjeistus painottaa ketteryyden oikein ymmärtämistä, jolloin sen onnistuminen on todennäköisempää. Kyseessä on filosofia, josta on tarkoitus ottaa omalle organisaatiolle tai projektille sopivat käytännöt käyttöön, ja soveltaa niitä itselle parhaalla mahdollisella tavalla. Tämän lisäksi ohjeistus pitää sisällään yleistettäviä ohjeistuksia, jotka parantavat lähes minkä tahansa organisaation sisäistä toimintaa, kuten avoin kommunikaatio, epäonnistumisen hyväksyminen ja jatkuva kehitys. Suunnittelijoiden kannalta ohjeistuksessa tärkeä suositus on resursoinnin optimointi siten, että suunnittelijoilla on aikaa tehdä tarpeeksi esitutkimusta, sekä luoda ja kommunikoida yhteinen visio koko tiimille. Toinen tärkeä suositus koskee suunnittelijoiden lisäksi koko muuta tiimiä – koko tiimin pitäisi olla mukana esiselvityksen tekemisessä, suunnittelussa ja testaamisessa, jotta yhteinen ymmärrys tarpeista, visiosta ja isosta kuvasta säilyy läpi koko projektin. Kolmas suositus on pitää iteraatiot lyhyinä, jotta palautetta saadaan toteutetuista ominaisuuksista ja tehdyistä suunnitelmista heti joko muulta tiimiltä, asiakkaalta tai loppukäyttäjiltä, ja korjaukset saadaan tehtyä tuotteeseen saman tien.

Tekemäni haastattelut vahvistivat omaa ajatustani siitä, että yrityksen oma toiminta toimii lähtökohtana käyttäjäkeskeisen suunnittelun ja ketteryyden onnistumiselle, sekä

näiden integroimiselle. Yritys voi sitoutua ja toteuttaa ketteryyttä monella eri abstraktiotasolla, kuten Leania käsittelevässä kappaleessa esitettiin. Ketteryys lähtee liikkeelle yrityksen tai organisaation arvoista ja päätyy työkaluihin ja toimintoihin (kuva 10). Ylemmät abstraktiotasot luovat puitteet niitä seuraaville tasoille. Esimerkiksi suunnittelijoiden ja kehittäjien epäonnistumisten hyväksyminen ja motivoiminen jatkuvaan parantamiseen on pitkälti kiinni siinä, miten yrityskulttuuri suhtautuu näihin asioihin. Yrityksen luomat puitteet luovat pohjan onnistuneelle käyttäjäkeskeiselle suunnittelulle, ketteryydelle ja näiden integroimiselle. Yritys voi myös parantaa onnistumismahdollisuuksia tekemällä asiakkaan kanssa sopimuksen, joka tukee kehitystiimin itseohjautuvuutta ja resursointia.

9.1 Pohdinta tutkimuksesta

Tutkimuksessa haastateltiin ainoastaan kuutta suomalaista ohjelmistoyritystä, joten otanta ei tulosten kattavuuden kannalta ollut kovin suuri. Suuremmalla otannalla tutkimustulosten vaihtelevuus olisi mahdollistanut useampien vaihtoehtoisten toimintatapojen löytymisen. Tutkimusta olisi parantanut myös työpaja, johon olisi osallistunut myös kehittäjiä, jolloin ketterää prosessia olisi voinut katsoa useammasta näkökulmasta. Nyt näkökulma oli rajattu ainoastaan suunnittelijoihin.

Haastattelukysymykset olivat avoimia kysymyksiä ja vastaajilla oli vapaus kertoa toimintatavoistaan ja menetelmistään haluamallaan tavalla. Tämä johti melko erilaisiin vastausmalleihin ja haastattelussa käsiteltäviin asioihin, vaikka vastaukset kaikkiin kysymyksiin saatiinkin. Haastateltavat osallistuivat haastatteluihin omalla työajallaan, kuten pidennetyillä lounastauolla, joten haastatteluaika oli toisinaan melko rajallinen, ja esimerkiksi asiakkaan toimintaan liittyviä kysymyksiä jouduttiin kiiruhtamaan läpi. Suunnittelijoiden vastaukset perustuivat heidän subjektiiviseen kokemukseen, ja heidän esimerkiksi kritisoidessaan kehittäjiä tai asiakasta, ei tutkimuksessa ollut mahdollisuutta kuulla toisen osapuolen näkemystä. Suunnittelijat vaikuttivat rehellisiltä haastatteluissa, paitsi yhdessä, jossa suunnittelijat eivät löytäneet omassa toimintatavassaan mitään kehitettävää, mikä voi kertoa myös sokeudesta omalle toimintatavalleen. Toki voi olla, että he ovat löytäneet itselleen sopivan toimintatavan toteuttaa integraatiota, mutta ketteryyden perustuessa jatkuvaan parantamiseen, voitaneen olettaa, että retrospektiiveille olisi tarvetta. Kaikki haastattelut nauhoitettiin, ja nauhoitteet käytiin läpi vielä haastattelujen jälkeen, jotta virheet muistiinpanoissa saatiin minimoitua.

Verkkoaineistoanalyysiä varten kävin läpi lähemmäs sata lähdettä, joista valikoitui diplomityöhön mukaan 23. Kriteereinä mukaanottoon olivat muun muassa kirjoittajan kokemus käyttäjäkeskeisyydestä ja/tai ketteryydestä, artikkelin yhteyteen lisätyt lähteet, ja sivusto, jossa artikkeli on julkaistu.

9.2 Vertailu edellisiin tutkimuksiin

Hyvät käytännöt käyttäjäkeskeisen suunnittelun integroimiseen ketteriin ohjelmistokehitysprosesseihin on näkökulmana melko harvinainen tutkimuksissa. Monet tutkimukset ovat lähteneet liikkeelle ongelmien tutkimisesta ja niiden ratkaisemisesta, joita voidaan toki kerätä yhteen hyviä käytäntöjä. Yleisimmin tutkimukset kuitenkin keskittyvät käyttäjäkeskeisen suunnittelijan roolin integroimiseksi ketterään kehitystiimiin [25].

Sy [39] esitteli prosessimallin, jolla yhdistetään suunnittelu ja kehitys, ja miten nollaiteraatio sisältää muun muassa esitutkimuksen. Yksittäisiä hyviä käytäntöjä ja toimintatapoja on käsitelty eri alojen tutkimuksista kuitenkin paljon. Muun muassa käyttäytymiseen ja johtamiseen liittyvistä tutkimuksista on johdettavissa ratkaisuja myös suunnittelu- ja kehitystyön integraatio-ongelmiin, kuten kommunikaatioon ja ryhmäytymiseen.

Vuonna 2015 Kuusinen julkaisi väitöskirjan [25], johon hän on koonnut BoB-viitekehyksen hyvistä käytännöistä, miten integraatiota voidaan edesauttaa yritysohjelmistojen kehitysprojekteissa. Viitekehys lähestyy integraatiota käyttäjäkeskeisten tehtävien ja koko tiimin kesken jaetun omistajuuden kautta. Väitöskirjassa käsitellään monia samoja ajatuksia, joita tässä diplomityössä on käsitelty, kuten tehtävien allokointi rooleille.

Tässä diplomityössä on koottu lisää käytäntöjä integraation onnistumiseksi. Diplomityössä tuodaan esille, että käyttäjäkeskeisen suunnittelutyön integroituminen ketterään ohjelmistoprojektiin lähtee liikkeelle toimittavan yrityksen organisaatio- ja johtamistoiminnoista, joiden tulee tukea sekä käyttäjäkeskeisen suunnittelun tärkeyttä, että integroitumista. Yrityskulttuuri ja yrityksen arvot luovat perustan koko kehitystiimin muun muassa kommunikaatiolle, yhteisvastuulle ja itseohjautuvuudelle. Tätä asiaa on käsitelty kirjallisuudessa, kuten *Tätä on lean* –kirjassa [31]. Diplomityössä fokusoidaan näkökulmaa ja sovelletaan tuloksia käyttäjäkeskeiseen suunnittelutyöhön ja sen integroitumiseen ketterään ohjelmistoprosessiin.

9.3 Tulevat tutkimukset ja jatkotutkimusaiheita

Yritykset kehittävät jatkuvasti omia menetelmiään käyttäjäkeskeisen suunnittelun ja kehityksen integroimiseksi. Näiden yksittäisten menetelmien yleistämistä muihin yrityksiin ja muille toimialoille olisi hyvä tutkia. Yksi tällainen on Google iteroimisprosessi *Google Design Sprint Process* [57], jota myös suomalaisissa yrityksissä on lähdetty kokeilemaan. Googlen iteroimisprosessin osalta voisi tutkia, millaisiin projekteihin se soveltuu.

Etenkin julkiset yritykset joutuvat kilpailuttamaan ohjelmisto- ja suunnitteluyrityksiä projekteihinsa. Tällöin projekteihin saattaa osallistua tekijöitä useista eri yrityksistä.

Tällaista asetelmaa olisi mielestäni hyvä tutkia hyvien käytäntöjen osalta, sillä asetelma on täysin erilainen verrattuna projekteihin, joissa koko toteutus tulee yhdeltä yritykseltä. Esimerkiksi yhteisvastuu toteutuksesta voi olla vaikeaa, kun jokainen yritys pyrkii suorittamaan oman työnsä virheettömästi asiakkaan silmissä. Usean yrityksen projekteissa yritykset ovat useimmiten myös kilpailijoita keskenään, ja tiimin jäseniin keskittyy erilaiset odotukset myös oman yrityksen osalta kilpailunäkökulmasta. Aihetta voisi tutkia niin suunnittelijoiden ja kehittäjien kuin asiakkaankin näkökulmasta.

Mielenkiintoinen tutkimusaihe olisi myös, miten kehitystiimin ulkopuoliset tekijät, kuten oman organisaation, asiakkaan ja eri sidosryhmien toiminta, vaikuttaa integroidun tiimin ja ketteryyden onnistumiseen. Miten näiden tekijöiden toiminta, vaatimukset ja osallistuminen eri tehtäviin vaikuttavat ketterään projektin onnistumiseen. Kuten haastatteluissakin tuli esille asiakkaan ketteryyden maturiteetissa voi olla paljon eroavaisuuksia, ja asiakkaan omat sisäiset prosessit saattavat hankaloittaa kehitystiimin toimintaa ja projektissa etenemistä.

9.4 Mitä tekisin toisin?

Tutkimussuunnitelman ja haastattelurungon suunnittelun aikana vasta tajusin kunnolla, kuinka suurta palaa olen haukkaamassa. Lisäksi aiheen laajuus kävi raskaaksi haastatteluanalyysin ja verkkoaineistoanalyysin aikana, koska vastauksia ja mielipiteitä kysymyksiin etsiessäni törmäsin todella usein vastaukseen ”*Riippuu projektista.*” Tämä vaikuttaa vahvasti siihen, millä kompetenssikokoonpanolla projektia tehdään, minkä tyyppinen projekti on kyseessä, millainen yritys on asiakkaana, mitä menetelmiä käytetään, mikä on aikataulu, millaisia aktiviteetteja projektin aikana voidaan tehdä ja niin edelleen. Minun olisi pitänyt tutkia aihetta enemmän etukäteen, ja rajata aihetta tarkemmin jo alussa. Aihetta olisi voinut tarkentaa tarkempaan katsontakantaa, keskittyä ainoastaan asiakasyhteistyöhön, tietyn roolin, kuten interaktiosuunnittelijan tehtäviin, tai kommunikaatiomenetelmiin, jolloin mukaan olisi mahtunut myös erilaiset työkalut. Nykyisillä tutkimuskysymyksillä saatiin kuitenkin hyvä läpileikkaus hyvistä käytännöistä, joita yrityksissä todellisuudessa käytetään haasteista huolimatta, ja ne voivat auttaa yrityksiä heidän omien käytäntöjen ja toimintatapojen kehityksessä.

LÄHTEET

[1] *SFS-EN ISO 9241-210*, Ihmisen ja järjestelmän vuorovaikutuksen ergonomia: Vuorovaikutteisten järjestelmien käyttäjäkeskeinen suunnittelu, Suomen standardoimisliitto, Helsinki, 2010, 48 s.

[2] *SFS-EN ISO 9241-11*, Näyttöpäätteillä tehtävän työn käyttäjäkeskeinen suunnitteluprosessi, Suomen standardoimisliitto, Helsinki, 1998.

[3] *SFS-EN ISO 13407*, Vuorovaikutteisten järjestelmien käyttäjäkeskeinen suunnitteluprosessi, Suomen standardoimisliitto, Helsinki, 1999, 34 s.

[4] P. Ahtinen, *Workshopin fasilitointi*, Opinnäytetyö, Ammattikorkeakoulu Savonia, Kuopio, 2014, 47 s.

[5] J. Armitage, *Are Agile Methods Good for Design?*, Interactions - Making scents: aromatic output for HCI, Vol. 11, No. 1, 2004, s. 14-23.

[6] M. Brhel, H. Meth, A. Maedche, C. Werder, *Exploring principles of user-centered agile software development: A literature review*, Information and Software Technology, Vol. 61, 2015, pp. 163-181.

[7] D. Brown, *Agile User Experience Design - A Practitioner's Guide to Making It Work.*, 1. painos. Morgan Kaufmann, San Francisco, CA, USA, 2013, 176 s.

[8] D.M. Brown, *Communicating Design: Developing Web Site Documentation for Design and Planning*, 2. painos. New Riders Press, San Francisco, US, 2011, 312 s.

[9] K. Collin, *Luovuus, oppiminen ja asiantuntijuus: koulutuksen ja työelämän näkökulmia*, 1. painos. WSOY, Helsinki, 2010, 253 s.

[10] A. Cooper, *The Inmates are Running the Asylum*, 1. painos. Sams Publishing, Indianapolis, USA, 1999, 288 s.

[11] T. Dingsøyr, S. Nerur, V. Balijepally, N. Moe, *A decade of agile methodologies: Towards explaining agile software development*, Journal of Systems and Software, Elsevier, Vol. 85, No. 6, 2012, s. 1213-1221.

[12] V.-P. Eloranta, K. Koskimies, *Sulava Scrum Survey Report*, 1. painos. Ohjelmistotekniikan laitos, Tampereen teknillinen yliopisto, Tampere, 2011, 27 s.

[13] J. Ferreira, J. Noble, R. Bibble, *Agile Development Iterations and UI Design*, Proceedings: Agile 2007, Washington, DC, USA, 13. - 17. elokuuta 2007, IEEE Computer Society, s. 50-58.

[14] D. Fox, J. Sillito, F. Maurer, *Agile Methods and User-Centered Design: How These Two Methodologies are Being Successfillu Integrated in Industry*, Proceedings: Agile 2008 Conference, Toronto, ON, Canada, 4. - 8. elokuuta 2008, IEEE Computer Society, s. 63-72.

[15] J. Gorhelf, J. Seiden, *Lean UX: Applying Lean Principles to Improve User Experience*, 1. painos. O'Reilly Media, Sebastopol, CA, USA, 2013, 152 s.

[16] D. Green, J.M. Pearson, *Development of a Web Site Usability Instrument Based on ISO 9241-11*, Journal of Computer Systems, Vol. 45, No. 1, 2006, s. 66-72.

[17] I. Haikala, T. Mikkonen, *Ohjelmistotuotannon käytännöt.*, 12. painos. Talentum Media Oy, Hämeenlinna, 2011, 242 s.

[18] K. Heikkilä, *Tiimit: avain uuden luomiseen*, 1. painos. Talentum Oyj, Helsinki, 2002, 445 s.

[19] S. Hyysalo, *Käyttäjätieto ja käyttäjätutkimuksen menetelmät*, 1. painos. Edita, Helsinki, 2006, 319 s.

[20] T. Jaakkola, J. Kataja, J. Liukkonen, *Ryhmä liikkeelle! Toiminnallisia harjoituksia ryhmän kehittämiseksi*, 1. painos. PS-Kustannus, Jyväskylä, 2011, 137 s.

[21] H. Janhunen, *Käyttäjätestaus*, Opinnäytetyö, Mikkelin ammattikorkeakoulu, Tietojen käsittely, 2013, 25 s.

[22] R. Jauhiainen, M. Eskola, *Ryhmäilmiö*, 1. painos. WSOY, Porvoo, 1994, 170 s.

[23] J. Kollman, H. Sharp, A. Blandford, *The importance of Identity and Vision to User Experience Design on Agile Projects*, Proceedings of the 2008 Agile Conference, Chicago, USA, 24. - 29. elokuuta 2009, IEEE Computer Society, Washington, DC, USA, s. 11-18.

[24] I. Kouri, *Lean-taskukirja*, 1. painos. Teknologiateollisuus ry, Helsinki, 2009, 39 s.

[25] K. Kuusinen, *Integrating UX Work in Agile Enterpise Software Development*, väitöskirja, Tampereen teknillinen yliopisto. Julkaisu 1339, 2015, 297 p.

[26] K. Kuusinen, T. Mikkonen, S. Pakarinen, *Agile User Experience Development in a Large Software Organization: Good Expertise but Limited Impact*, Human-Centered Software Engineering, 4th International Conference, HCSE 2012, Toulouse, France, 29. - 31. lokakuuta 2012, Springer Verlag, Berlin, Heidelberg, s. 94-111.

[27] W. Lidwell, K. Holden, J. Butler, *Universal Principles of Design*, 1. painos. Rockport Publishers, Lontoo, UK, 2010, 182 s.

[28] J.K. Liker, *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer*, 1. painos. McGraw-Hill Education, 2004, 330 s.

[29] D. Lu, Japan Management Institute, *Kanban Just-in Time at Toyota: Management Begins at the Workplace*, 1. painos. Productivity Press, 1986, 186 s.

[30] J. Metsämuuronen, *Tutkimuksen tekemisen perusteet ihmistieteissä*, 4. painos. International Methelp Oy, Helsinki, 2009, 1632 s.

[31] N. Modig, P. Åhlström, *Tätä on Lean: Ratkaisu tehokkuusparadoksiin*, 2. painos. Rheologica, Tukholma, Ruotsi, 2013, 167 s.

[32] J. Nielsen, *Usability Engineering*, 1. painos. Morgan Kaufmann, San Francisco, CA, USA, 1993, 362 s.

[33] T. Ohno, *Toyota Production System: Beyond Large-Scale Production*, 1. painos. Production Press, Oregon, USA, 1988, 143 s.

[34] Y. Rogers, H. Sharp, J. Preece, *Interaction Design: Beyond Human-Computer Interaction*, 3. painos. John Wiley & Sons Ltd., West Sussex, England, 2011, 585 s.

[35] W.W. Royce, *Managing the development of large software systems: concepts and techniques*, ICSE '87 Proceedings of the 9th international conference on Software Engineering, Monterey, CA, USA, 30. maaliskuuta - 2. huhtikuuta 1987, IEEE Computer Society Press, Los Alamitos, CA, USA, s. 328-338.

[36] D. Salah, R. Paige, P. Cairns, *A Systematic Literature Review on Agile Development Processes and User Centred Design Integration*, Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, Lontoo, UK, 13. - 14. toukokuuta 2014, ACM, New York, USA.

[37] A. Seffah, E. Metsker, *The Obstacles and Myths of Usability and Software Engineering*, Communications of the ACM, Vol. 47, No. 12, 2004, s. 71-76.

[38] Summa, T.; Tuominen, K., *Fasilitaattorin työkirja - Menetelmiä sujuvaan ryhmätyöskentelyyn*, Kehitysyhteistyön palvelukeskus Kepa ry, 2009, 66 s.

[39] D. Sy, *Adapting Usability Investigation for Agile User-Centered Design*, Journal of Usability Studies, Vol. 2, No. 3, 2007, s. 112-132.

[40] D. Sy, L. Miller, *Optimizing Agile User-centered Design*, Proceedings of the Conference on Human Factors in Computing Systems, 5. - 10. huhtikuuta 2008, ACM Press, New York, USA, s. 3897-3900.

[41] K. Väänänen-Vainio-Mattila, V. Roto, M. Hassenzahl, *Towards Practical User Experience Evaluation Methods*, Proceedings of the 5th COST294-MAUSE Open Workshop on Meaningful Measures: Valid Useful User Experience Measurement (VUUM 2008), 18.06.2008, Institute of Research in Informatics of Toulouse (IRIT), Toulouse, France, s. 19-22.

[42] J.P. Womack, D.T. Jones, *Lean Thinking*, 2. painos. Simon & Schuster, Inc., USA, 1994, 397 s.

[43] J.P. Womack, D.T. Jones, D. Roos, *Machine That Changed the Wold*, 1. painos. Scribner, 1990, 336 s.

VERKKOLÄHTEET

[44] *Agile Innovation Solutions: Product Owner Faces Two Directions*, Kenneth S. R. and Innolution LLC, WWW. Saatavilla (viitattu 29.05.2016): http://www.innolution.com/val/detail/product-owner-faces-two-directions.

[45] *Agile Manifesto - documentation*, LinkedIn Corporation, WWW. Saatavilla (viitattu 28.09.2016): https://www.linkedin.com/groups/37631/37631-6022651040384512003.

[46] *Agile Modeling*, Ambysoft Inc., WWW. Saatavilla (viitattu 13.12.2016): http://agilemodeling.com.

[47] *Balsamiq*, Balsamiq Studios, LLC, WWW. Saatavilla (viitattu 09.06.2016): https://balsamiq.com.

[48] *Business Dictionary*, WebFinance, Inc., WWW. Saatavilla (viitattu 05.06.2016): http://www.businessdictionary.com/definition/customer-experience.html.

[49] *Code Camp for Girls*, Frantic, WWW. Saatavilla (viitattu 10.06.2016): https://www.frantic.com/codecamp/termipeli_codecamp.pdf.

[50] *Finder, Fonecta, WWW. Saatavilla (viitattu 10.06.2016):* https://www.finder.fi.

[51] *Framer*, Motif Tools BV, WWW. Saatavilla (viitattu 09.06.2016): http://framerjs.com.

[52] *If Agile is a continuous improvement and transformation! Why we are calling Agile is failing some times?*, LinkedIn Corporation, WWW. Saatavilla (viitattu 06.09.2016): https://www.linkedin.com/groups/37631/37631-6036421278959755268.

[53] *InVision*, InVision, WWW. Saatavilla (viitattu 09.06.2016): https://www.invisionapp.com.

[54] *Kielitoimiston sanakirja*, Kotimaisten kielten keskus ja Kielikone Oy, WWW. Saatavilla (viitattu 29.05.2916): http://www.kielitoimistonsanakirja.fi.

[55] *Lean-sanasto*, Leaniksi, WWW. Saatavilla (viitattu 29.05.2016): http://leaniksi.fi/lean-sanasto/.

[56] *LinkedIn*, LinkedIn Corporation, WWW. Saatavilla (viitattu 09.06.2916): https://www.linkedin.com.

[57] *Make Your UX Design Process Agile Using Google's Methodology*. Interaction Design Foundation, 2016, WWW. Saatavilla (viitattu 29.09.2016): https://www.interaction-design.org/literature/article/make-your-ux-design-process-agile-using-google-s-methodology

[58] *Manifesto for Agile Software Development*, Scrum.org, WWW. Saatavilla (viitattu 28.05.2016): http://www.agilemanifesto.org/iso/en/.

[59] *Medium*, Medium Corporation, WWW. Available (accessed 13.12.2016): https://medium.com.

[60] *Mitä on asiakaskokemus?*, Asiakaskokemus.fi, WWW. Saatavilla (viitattu 05.06.2016): http://www.asiakaskokemus.fi/2011/01/mita-on-asiakaskokemus/.

[61] *Nielsen Norman Group*, WWW. Saatavilla (viitattu 13.12.2016): https://www.nngroup.com.

[62] *Sketch*, Bohemian Coding, WWW. Saatavilla (viitattu 09.06.2016): http://www.sketchapp.com.

[63] *Suomisanakirja*, SuomiSanakirja.fi, WWW. Saatavilla (viitattu 07.06.2016): http://www.suomisanakirja.fi.

[64] *The Agile Manifesto*, Agile Alliance, WWW. Saatavilla (viitattu 09.06.2016): https://www.agilealliance.org/agile101/the-agile-manifesto/.

[65] *UXmatters*, Pabini Gabriel-Petit, WWW. Saatavilla (viitattu 13.12.2016): http://www.uxmatters.com.

[66] *While Agile proposes Less Documentation=> Should "Key Techincal Decisions" still be documented?*, LinkedIn Corporation, WWW. Saatavilla (viitattu 15.11.2016): https://www.linkedin.com/groups/37631/37631-6044548776507490308.

[67] *Working software over comprehesive documentation?*, LinkedIn Corporation, WWW. Saatavilla (viitattu 06.10.2016): https://www.linkedin.com/groups/37631/37631-5999864918701064194.

[68] S. W. Ambler, *User Stories: An Agile Introduction*, Agile Modelling, WWW. Saatavilla (viitattu 09.06.2016): http://www.agilemodeling.com/artifacts/userStory.htm.

[69] C. Cousins, *Why Designers and Web Developers Must Work Together*, Design-modo, WWW. Saatavilla (viitattu 03.12.2016): https://designmodo.com/designers-developers-work/.

[70] A. Cowan, *The Enterprise Software Playbook - 6 Steps to Better Deployments*, Alex Cowan, WWW. Saatavilla (viitattu 09.06.2016): http://www.alexandercowan.com/salesforce-playbook-6-steps-better-deployments/.

[71] B. Crothers, *Storyboarding & UX - part 1: an introduction*, Johnny Holland, WWW. Saatavilla (viitattu 09.06.2016): http://johnnyholland.org/2011/10/storyboarding-ux-part-1-an-introduction/.

[72] R. Fotolescu, *Why UX Design Should be Aligned with Development Sprints*, Adobe Systems Incorporated, WWW. Saatavilla (viitattu 04.12.2016): https://blogs.adobe.com/creativecloud/why-ux-design-should-be-aligned-with-development-sprints/.

[73] J. - C. Grosjean, *Agile UX Practices*, Agile UX, WWW. Saatavilla (viitattu 03.12.2016): http://www.agile-ux.com/2010/11/21/agile-ux-in-practice/.

[74] J. Harris, *Agile UX*, UXmatters, WWW. Saatavilla (viitattu 04.12.2016): http://www.uxmatters.com/mt/archives/2015/09/agile-ux.php.

[75] G. Heller, *Scrum Visualized*, Gregory Heller, WWW. Saatavilla (viitattu 09.06.2016): http://home.gregoryheller.com/scrum-diagram.

[76] R. Hellström, *Ketterä Keskiviikko - Agile teemana*, Agile HR, WWW. Saatavilla (viitattu 06.09.2016): http://www.agile4hr.com/2015/09/03/kettera-keskiviikko-agile-teemana/.

[77] E. Ilama, *Creating Personas*, UX Booth, WWW. Saatavilla (viitattu 09.06.2016): http://www.uxbooth.com/articles/creating-personas/.

[78] J. Kiminki, *Mikä on Agile? Ketteryyden perusteet*, LinkedIn Corporation, WWW. Saatavilla (viitattu 29.09.2016): http://www.slideshare.net/jkiminki/ketteryyden-perusteet-.

[79] L. Lekman, *Mikä ihmeen Kanban?*, Lekman Consulting, WWW. Saatavilla (viitattu 08.06.2016): http://lekman.fi/2009/09/26/mika-ihmeen-kanban/.

[80] P. Leppäniemi, *Ketterät menetelmät: Scrum*, WWW. Saatavilla (viitattu 09.06.2016): http://ohjelmistotuotanto.panuleppaniemi.com/agile-scrum/.

[81] H. Loranger, *Top 10 Tips for UX Success From Agile Practitioners*, Nielsen Norman Group, WWW. Saatavilla (viitattu 04.12.2016): https://www.nngroup.com/articles/ux-success-agi-le/?utm_content=buffera8367&utm_medium=social&utm_source=linkedin.com&utm_campaign=buffer.

[82] C. Naji, *How Lean UX builds better products: Q&A with Jeff Gothelf*, JustInMind, WWW. Saatavilla (viitattu 04.12.2016): https://www.justinmind.com/blog/how-lean-ux-builds-better-products-qa-with-jeff-gothelf/.

[83] J. Nielsen, *Agile Development Projects and Usability*, Nielsen Norman Group, WWW. Saatavilla (viitattu 04.12.2016): https://www.nngroup.com/articles/agile-development-and-usability/.

[84] J. Nielsen, *Agile User Experience Projects* , Nielsen Norman Group, WWW. Saatavilla (viitattu 04.12.2016): https://www.nngroup.com/articles/agile-user-experience-projects/.

[85] S. Ovaska, A. Aula, P. Majaranta, *Käytettävyystutkimuksen menetelmät*, Tietojenkäsittelytieteiden laitos, Tampereen teknillinen yliopisto, WWW. Saatavilla (viitattu 29.05.2016): http://docplayer.fi/10276342-Saila-ovaska-anne-aula-ja-paivi-majaranta-toim.html.

[86] A. Ramsay, *Designing Ahead: The Good, The Bad, and The Ugly*, Coder Chronicles, WWW. Saatavilla (viitattu 03.12.2016): http://coderchronicles.org/2010/08/22/designing-ahead-the-good-the-bad-and-the-ugly/.

[87] A. Rosenstein, *The UX Professionals' Guide to Working with Agile Scrum Teams* , Boxes and Arrows, WWW. Saatavilla (viitattu 03.12.2016): http://boxesandarrows.com/the-ux-professionals-guide-to-working-with-agile-scrum-teams/.

[88] A. Saaranen-Kauppinen, A. Puusniekka, A. Kuula, R. Rissanen, I. Karvinen, KvaliMOTV - menetelmänopetuksen tietovarasto. Teema haastattelu, Yhteiskuntatieteellinen tietoarkisto, Tampereen yliopisto, WWW. Saatavilla (viitattu 29.05.2016): http://www.fsd.uta.fi/fi/julkaisut/motv_pdf/KvaliMOTV.pdf.

[89] K. Schrabber, J. Sutherland, *Scrum Guide - Scrumin määritelmä ja pelisäännöt*, Scrum.Org & ScrumInc., WWW. Saatavilla (viitattu 09.06.2016): http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-FI.pdf.

[90] J. M. Six, *Integrating UX into Agile Development*, UXmatters, WWW. Saatavilla (viitattu 04.12.2016): http://www.uxmatters.com/mt/archives/2011/04/integrating-ux-into-agile-development.php.

[91] J. M. Six, *Agile User Experience Design*, UXmatters, WWW. Saatavilla (viitattu 04.12.2016): http://www.uxmatters.com/mt/archives/2012/04/agile-user-experience-design.php.

[92] N. Van Weerdenburg, How Lean UX Fixes Common Agile Challenges (AgileXD - April, 2016), YouTube, Rangle.io, WWW. Saatavilla (viitattu 28.09.2016): https://www.youtube.com/watch?v=TP8jKJOd1cY.

[93] F. van Wingerde, *Agile and User Experience: The Real Problem Is Handing Over*, Medium Corporation, WWW. Saatavilla (viitattu 08.12.2016): https://medium.com/@fj/agile-and-user-experience-the-real-problem-is-handing-over-9f3c47fef91#.ak1znkkxz.

[94] V. S. Virsta, *Haastattelutavat, Strukturoitu haastattelu*, Tilastokeskus ja Helsingin ammattikorkeakoulu Stadia, WWW. Saatavilla (viitattu 29.05.2016): https://www.stat.fi/virsta/tkeruu/04/01/.

[95] J. Winter, *Everything I've learned about Lean UX: AMA with Jeff Gothelf*, UserTesting, WWW. Saatavilla (viitattu 04.12.2016): https://www.usertesting.com/blog/2016/07/08/jeff-gothelf/.

[96] A. Witteman, *UX & Agile – UX Research Amsterdam meetup – 04022016*, LinkedIn Corporation, WWW. Saatavilla (viitattu 04.12.2016): http://www.slideshare.net/icemobile/ux-agile-ux-research-amsterdam-meetup-04022016-57913171.

LIITE A: HAASTATTELURUNKO

TAUSTATIEDOT

- Firman perustamisvuosi
- Firman virallinen toimialaluokitus
- Firman henkilöstöluokka
- Firman liikevaihto vuodessa
- Suunnittelijoiden kompetenssialueet
- Käytetyt Agile-menetelmät

AGILE-PROJEKTIN TIIMI

- Kuvaile Agile-kehitystiimin yleisintä kokoonpanoa.
    a. Montako suunnittelijaa kehitystiimissä keskimäärin on?
    b. Ovatko suunnittelijat Agile-tiimissä oma sisäinen tiiminsä vai tekijöitä muiden joukossa?
- Onko suunnittelijat mukana koko Agile-projektin ajan?
- Ovatko suunnittelijat projektikohtaisia, vai työskenteleekö suunnittelija useassa Agile-tiimissä samanaikaisesti?
- Onko projektibudjetista erotettu suunnittelijoille oma osansa, vai jakaako kehitystiimi budjetin itseohjautuvasti keskenään?

SUUNNITTELIJOIDEN ROOLIT JA TEHTÄVÄT

- Kuka tai mikä määrää, mitä suunnittelutehtäviä projektissa tehdään?
- Kuka jakaa roolit ja tehtävät? Vai onko rooli- ja tehtävänjako täysin itseohjautuvaa tiimin sisällä?
- Miten roolit, tehtävät ja vastuualueet kommunikoidaan tiimin kesken?
- Miten suunnittelijat näkevät tämän asetelman?

AGILE-TIIMIN KOMMUNIKOINTI

- Millaista on hyvä kommunikointi tiimissä?
- Mitkä asiat vaikuttavat hyvään kommunikointiin?
- Mitä työvälineitä käytetään kommunikointiin?
- Miten kommunikointia voidaan vielä parantaa?
- Osallistuvatko kaikki tiimin jäsenet asiakaspalavereihin?
- Onko tiimillä omia sisäisiä palavereja? Millaisia?
- Työskenteleekö tiimi samassa tilassa?
- Miten koko tiimi pysyy perässä projektin kokonaiskuvassa ja suunnittelijoiden työstä? Miten selkeä kokonaiskuva suunnittelijoiden mielestä on kehittäjille?

TEHTÄVÄ: PROSESSIN KUVAAMINEN

- Piirrä paperille prosessimalli / iteraatiosykli omasta näkökulmastasi.

    a. Voit piirtää useamman kuvan erilaisista menetelmistä ja tilanteista.
- Mihin vaiheisiin suunnittelija osallistuu?
- Missä vaiheissa on suunnittelijalla suurin työmäärä ja missä pienin?

ESISELVITYS

- Kuvaile esiselvitystyötä suunnittelijan näkökulmasta.
- Onko esiselvitys kiinteä osa projektityötä (myös budjetoinnin kannalta)?
- Mitä tietoa asiakas tarjoaa valmiina?
- Mitä esiselvityksiä tehdään ennen varsinaisen käyttäjäkeskeisen suunnittelun aloittamista?
- Mikä on pienin esiselvitys mitä pitää tehdä ja mitä pitää vähintään tietää?
- Keitä esiselvitysprosessiin osallistuu?
- Miten tulokset käydään läpi, dokumentoidaan ja kommunikoidaan eteenpäin?
- Miten esiselvitys vaikuttaa projektin vaatimuksiin ja kehitysjonoon?
- Arvioi esiselvitykseen käytettävää aikaa?
- Onko esiselvityksestä oikeasti hyötyä? Mitä?
- Onko jostain esiselvitystehtävistä ajan saatossa luovuttu? Miksi?

KÄYTTÄJÄKESKEINEN SUUNNITTELU

- Mikä on tarpeellista suunnittelua, ja mistä on ajan saatossa luovuttu?
- Kuvaile ja arvioi suunnitteluprosessia.
- Miten suunnittelukokonaisuus paloitellaan iteratiivisessa suunnittelumallissa?
- Millä tarkkuustasolla paloittelu tehdään?
- Ketkä paloitteluun osallistuvat?
- Miten suunnittelutehtävät jaetaan koko kehitystiimin kesken?
- Arvioi käyttäjäkeskeisen suunnittelun työmäärää eri vaiheissa prosessia.
- Missä vaiheessa ja miksi pullonkauloja syntyy? Miten tätä voisi helpottaa?
- Mitä minimisuunnittelutyö, jotta kehitystyö voidaan aloittaa?
- Mistä suunnittelutehtävistä voitaisiin luopua?
- Miten suunnittelupäätökset kommunikoidaan kehittäjille?
- Tarvitaanko suunnitelmille asiakkaan hyväksyntä ennen toteuttamista?
- Mitä suunnittelusta dokumentoidaan?

SUUNNITTELU- JA AGILEITERAATIOIDEN INTEGROIMINEN

- Miten suunnittelu- ja kehitystyö jaksotetaan ja sovitetaan yhteen?
    a. Arvioi yhteensovittamisen toimivuutta.
    b. Mitkä menetelmät ja toimintatavat ovat tässä ratkaisevassa asemassa?
- Miten integraatiota voitaisiin parantaa?
- Arvioiko tai tarkistaako suunnittelija aikaan saadun toteutuksen ennen sen julkaisemista?

KÄYTETTÄVYYSTESTAUS

- Testaanko kehitetyt ratkaisut loppukäyttäjille ennen julkaisua?

- Missä vaiheessa käytettävyystestausta tehdään?
- Ketkä osallistuvat?
- Mitä testausmenetelmiä käytetään?
- Arvioi käytettävyystestaamisen tarkoituksenmukaisuutta ja siitä saatavaa lisäarvoa.

MUUTOSHALLINTA

- Kuvaile ja arvioi muutosten hallintaprosessia suunnittelijan näkökulmasta.
- Mistä/Keneltä muutostarpeet yleensä tulevat?
- Arvioi, mitkä ovat "hyväksyttäviä" ja mitkä ikäviä muutoksia?
- Miten muutokset otetaan mukaan Agile-prosessiin?
- Miten priorisoidaan?
- Kuka priorisoi?
- Miten muutoshallinta on otettu huomioon käyttäjäkeskeisen suunnittelun resursointiarvioinneissa? Varaudutaanko muutoksiin etukäteen?

ASIAKKAAT JA AGILE UX -PROJEKTI

- Arvioi asiakkaiden osallistumista ketteriin projekteihin.
- Miten kehitystiimi osallistaa asiakasta projektiin?
- Miten oma-aloitteinen ja kiinnostunut asiakkaan on hyvä olla?
- Arvioi optimaalista asiakkaan toimintaa Agile UX -projektissa.